Une approche ontologique pour l'interopérabilité et la composition automatique de services Web : application en astrophysique

Thierry Louge

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Systèmes Informatiques

Présentée et soutenue par :

M. THIERRY LOUGE

le mercredi 5 juillet 2017

Titre :

Une approche ontologique pour l'interopérabilité et la composition automatique de services Web : application en astrophysique.

Ecole doctorale :

Systèmes (Systèmes)

Unité de recherche :

Laboratoire Génie de Production de l'ENIT (E.N.I.T-L.G.P.)

Directeur(s) de Thèse :

M. JÜRGEN KNODLSEDER

M. BERNARD ARCHIMEDE

Rapporteurs :

M. CHIRINE GHEDIRA GUEGAN, UNIVERSITE LYON 3

M. VINCENT CHAPURLAT, ENSTIMA ALES

Membre(s) du jury :

M. CHIHAB HANACHI, UNIVERSITE TOULOUSE 1, Président

M. BERNARD ARCHIMEDE, ECOLE NATIONALE D'INGENIEURS DE TARBES, Membre

M. ERIC JOSSELIN, UNIVERSITE MONTPELLIER 2, Membre

MI. JÜRGEN KNODLSEDER, OBSERVATOIRE MIDI PYRENEES, Membre

Mme MIREILLE LOUYS, UNIVERSITE STRASBOURG, Membre

M. MOHAMED HEDI KARRAY, ECOLE NATIONALE D'INGENIEURS DE TARBES, Membre

Remerciements

Le moment d'écrire ces remerciements conclut plus de quarante mois de travaux et sonne, je l'espère, le départ d'années enthousiasmantes dans un travail de recherche passionnant. Je tiens à remercier en premier lieu mes trois encadrants.

Bernard ARCHIMEDE, bien sûr pour avoir accepté de diriger ce travail de thèse mais aussi pour le soutien solide qu'il m'a apporté depuis le premier jour de son enseignement en l'an 2000 au CNAM. Ce travail n'aurait pas été possible sans lui.

Jürgen KNODLSEDER pour l'engagement qu'il a pris dans l'encadrement d'un sujet interdisciplinaire peu commun au sein de l'IRAP, ainsi que le temps qu'il y a consacré malgré des responsabilités importantes au sein du consortium CTA.

Mohamed Hedi KARRAY dont j'ai eu l'honneur d'être le premier doctorant, pour l'énergie et l'enthousiasme communicatifs qu'il sait partager comme sa connaissance précise du domaine des ontologies. Ces conseils ont toujours été de première importance.

Je tiens aussi à remercier Sylvie BRAU-NOGUE pour la confiance qu'elle m'a accordée en me mettant en relation avec Jürgen KNODLSEDER, anticipant une collaboration fructueuse.

Mes remerciements s'adressent également avec force aux membres du jury, Prof. Chirine GHEDIRA-GUEGAN, Prof. Vincent CHAPURLAT, Dr. Mireille LOUYS, Prof. Chihab HANACHI et Dr. Eric JOSSELIN. Leur relecture détaillée de ce manuscrit, les remarques et suggestions qu'ils ont émises ont évidemment beaucoup contribué à améliorer la qualité de ce document ; mais elles constituent surtout des recommandations pour mes travaux à venir.

Un travail de thèse nécessite un engagement tel qu'il ne peut se dérouler convenablement que dans un environnement de travail convivial, dont j'ai pu bénéficier à l'Observatoire Midi-Pyrénées notamment en côtoyant Christophe MONTHEIL, ingénieur d'études au Télescope Bernard Lyot et Pascal PETIT, responsable scientifique de la base de données POLARBASE.

L'équipe DIDS du laboratoire LGP de l'ENIT procure aux doctorants qu'elle encadre un accueil exemplaire et des échanges toujours stimulants. Les chercheurs titulaires, comme mes collègues doctorants, ont tous tenu un rôle important durant cette thèse. Qu'ils en soient remerciés et pour n'en citer que quelques-uns Philippe FILLATREAU, Pascale CHIRON, Da XU, Justin MOSKOLAI…

Enfin, au moment où cette aventure s'achève, je remercie pour leur soutien indéfectible et l'affection que nous partageons ma famille, mes amis, les neveux, nièces et filleules qui sont apparus pendant cette période ou peu avant… Et aux premières loges des heures passées devant ma table de travail, ma compagne Stéphanie.

Résumé

Dans le but d'exploiter au mieux les grandes masses de données hétérogènes produites par les instruments scientifiques modernes de l'astrophysique, les scientifiques ont développé le concept d'Observatoire Virtuel (OV). Il s'agit d'une architecture orientée services, qui a pour objectif de faciliter l'identification et l'interopérabilité des données astrophysiques. Malgré le développement et les avancées permises par l'OV dans l'exploitation de ces données, certains objectifs sont partiellement atteints notamment l'interopérabilité, la sélection de services et l'identification de services connexes, etc. Par ailleurs, l'ergonomie des outils à la disposition de l'utilisateur final reste perfectible. De même l'utilisation actuelle des ressources de l'OV, s'appuyant sur des compétences humaines, gagnerait à être automatisée. Les services de données astrophysiques n'étant pas tous inscrits dans l'OV, il serait aussi souhaitable pour permettre une utilisation plus large de ces outils, qu'ils s'appuient également sur des services disponibles en-dehors de l'OV.

En vue d'automatiser l'utilisation des ressources en ligne, les sciences de l'information travaillent depuis 2001 à l'élaboration du Web sémantique. Cette évolution apporte au Web des capacités de raisonnement automatiques, basées sur des algorithmes utilisant une nouvelle forme de description des contenus. Cette nouvelle forme de description sémantique se trouve exprimée dans des représentations informatiques appelées ontologies. Malheureusement, les méthodes actuelles d'élaboration du Web sémantique ne sont pas complètement compatibles avec les services OV qui utilisent des modèles de données, des formats et des protocoles d'accès aux services qui s'éloignent de ceux rencontrés habituellement dans les sciences de l'information.

Dans ce contexte, cette thèse décrit une méthodologie générique de composition de services sans état, basée sur la description des services par une ontologie dont la définition est proposée dans ce document. Cette ontologie représente aussi bien des services Web que des services non accessibles par le Web. Elle prend en compte certaines spécificités qui peuvent être rencontrées dans les infrastructures de services préexistantes. L'enrichissement de l'ontologie par des concepts issus de domaines d'application spécifiques pour lesquels il n'existe que peu de représentations ontologiques est également pris en compte. La population de cette ontologie, par des services éventuellement éloignés des standards utilisés habituellement dans les sciences de l'information, est aussi traitée. La méthodologie a été appliquée avec succès dans le cadre de l'astrophysique, et a permis de développer une application Web permettant la composition automatique de services utilisable par un public non averti.

Mots-clés : Ontologies, services Web, interopérabilité, composition automatique de services, web sémantique, observatoire virtuel, astrophysique.

Abstract

Scientists have developed the Virtual Observatory (VO) concept in order to make the most of the large masses of heterogeneous data produced by the modern scientific instruments of astrophysics. It is a service-oriented architecture, aiming to facilitate the identification and interoperability of astrophysical data. Despite the development and advances made by VO in the exploitation of these data, some objectives are partially such as interoperability, service selection and identification of related services, etc. In addition, the ergonomics of the tools available to the end user can be improved. Similarly, the current use of VO resources, based on human skills, would benefit from being automated. As not all the astrophysical data services are included in the VO, it would also be desirable to allow a wider use of these tools, as they also rely on services available outside the VO.

In order to automate the use of online resources, information sciences have been working since 2001 on the development of the Semantic Web. This evolution provides the Web with automatic reasoning abilities, based on algorithms using a new form of content description. This new form of semantic description is expressed in computer representations called ontologies. Unfortunately, the current semantic Web development methods are not fully compatible with VO services that use data models, formats and protocols for accessing services that differ from those typically encountered in information sciences.

In this context, this thesis describes a generic methodology for the composition of stateless services, based on the description of services by a global ontology, the definition of which is proposed in this document. This ontology represents both Web services and services that are not accessible via the Web. It takes into account certain specificities that may be encountered in pre-existing service infrastructures. The enrichment of the ontology by concepts derived from specific fields of application for which there are only a few ontological representations is also taken into account. The population of this ontology, by services possibly distant from the standards usually used in the information sciences, is also treated. The methodology was applied successfully in the framework of astrophysics, and allowed to develop a Web application allowing the automatic composition of services usable by an uninformed public.

Keywords: Ontologies, Web services, interoperability, automatic service composition, semantic web, virtual observatory, astrophysics.

Sommaire

Liste des figures

Liste des tableaux

Liste des algorithmes

Liste des sigles, abréviations et acronymes

ASON : Astrophysical Services ONtology

ASON-ASTRO-THEM : ASON thematic module for astrophysics

CTA : Cherenkov Telescope array

DAML: DARPA Agent Markup Language

DAML-S : Semantic markup for Web services

DCI : Distributed Computer Infrastructure

EI: Extraction d'Informations

GEOS : GEneric Ontology for Services

IPDA: International Planetary Data Alliance

IVOA: International Virtual Observatory Alliance

KIF: Knowledge Interchange Format

ML: Machine Learning

NLP: Natural Language Processing

OV : Observatoire Virtuel

OWL : Web Ontology Language

OWL-S: Web Ontology Language for Services

RDF: Resource Description Framework

SGN : Science Gateway

SPARQL: SPARQL Protocol and RDF Query Language

SQWRL : Semantic Query-enhanced Web Rule Language

SSA (ou SSAP): Simple Spectrum Access (Protocol)

SWRL : Semantic Web Rule Language

SWS : Services Web Sémantiques

UCD: Unified Content Descriptor

UDDI: Universal Description Discovery and Integration

URI: Uniform Resource Identifier

URL: Uniform Resource Locator

UTYPE: Identifier of a data-model element in IVOA standards

VAMDC: Virtual Atomic and Molecular Data Center

W3C: World Wide Web Consortium

WSDL: Web Services Description Language

WSMO: Web Services Modeling Ontology

XML: eXtensible Markup Language

Introduction

Les instruments scientifiques modernes produisent des quantités de données numériques importantes, dont le volume et la multiplicité sont en croissance constante. Disposer de moyens de stockage et de sauvegarde suffisants pour assurer la pérennité des connaissances contenues dans cette masse d'informations est une problématique abordée par plusieurs axes de recherche en informatique.

Ces moyens de stockage et de sauvegarde doivent être associés à des capacités suffisantes d'accès et de fouille à l'intérieur de ces grandes quantités d'informations, pour en valoriser le contenu. La valorisation de ces données comporte les deux aspects, que sont la capacité d'extraire des résultats des analyses de ces données et l'établissement de critères de choix permettant de sélectionner les résultats à exploiter les plus prometteurs.

Plusieurs champs de recherche en informatique s'intéressent aux problématiques d'accès et de fouille de données adaptées à de grandes masses d'informations. L'émergence de connaissances cachées dans ces volumes importants se heurte à de nombreux problèmes de fragmentation et d'hétérogénéité technique et sémantique. Ces problèmes sont causés par les différences de nature (données « brutes » ou traitées, synthétiques ou observées…), de thèmes (astéro-sismologie, études des plasmas, cosmologie…), d'usages (stockage, archivage, traitement scientifique…) et de destinations (bases de données, publications…) pour lesquelles ces informations sont produites.

A l'heure où l'interdisciplinarité se développe à l'intérieur du monde scientifique, les données recueillies et produites restent la plupart du temps cloisonnées dans leur domaine d'origine. Ce cloisonnement s'explique principalement par des spécificités instrumentales ou scientifiques, qui amènent les fournisseurs à décrire leurs données dans le but de favoriser l'usage auquel elles sont destinées en premier lieu. En France et au sein du CNRS, le GdR MaDICS[1] s'intéresse à ces problématiques, comme la *Research Data Alliance*[2] le fait au niveau international. Le but de ces initiatives est le même : Permettre une recherche scientifique centrée sur des données massivement partagées, interopérables et cohérentes.

Le cloisonnement constaté entre les données de différentes disciplines scientifiques se retrouve non seulement au sein des sous-disciplines d'un même domaine, mais également entre des données d'une même sous-discipline. Les raisons de ce cloisonnement des données, y compris pour des sujets

[1] http://www.madics.fr/presentationmadics/

[2] https://rd-alliance.org/about-rda/who-rda.html

scientifiques proches, sont les mêmes que celles qui conduisent à l'hétérogénéité de plus haut niveau : la spécificité des données, les usages de la discipline ou sous-discipline considérée et l'utilisation principale envisagée pour les données produites. Le travail de cette thèse s'inscrit dans le cadre de l'interopérabilité, de la recherche et de la combinaison de services Son but est de faciliter la collaboration entre des services d'origines différentes, afin de permettre aux scientifiques d'utiliser des données dont ils pourraient ignorer l'existence, la définition ou le contenu. La recherche d'observations ou de résultats utilisables pour un problème particulier doit s'accompagner d'une sélection et d'une combinaison automatique de services fournisseurs de données, problématique également adressée dans ce manuscrit.

1. Contexte applicatif

En astrophysique comme dans d'autres domaines scientifiques, techniques ou industriels, le volume de données collecté par les instruments d'observation ou de mesures modernes présente une croissance exponentielle. De plus, ces données sont issues d'instruments de plus en plus complexe et coûteux. Les projets d'instrumentation importants en astrophysique comme « Cherenkov Telescope Array » (CTA)[3] ou EUCLID[4] par exemple, sont gérés par des consortia internationaux et mobilisent d'importants moyens financiers. Pour CTA, 32 pays forment le consortium et le projet est financé à hauteur d'environ 400 M€.

Les informations récoltées par ces instruments ont donc un coût financier important, le partage et l'interopérabilité des données qu'ils produisent est en conséquence un enjeu majeur pour le succès de ce type de projets. Devant ce constat, une organisation internationale regroupant des chercheurs, des techniciens et des ingénieurs a été créée au début du 21ème siècle. Cette organisation, *l'International Virtual Observatory Alliance* (IVOA)[5] met en place des outils, des protocoles et des modèles de données pour l'astrophysique visant à améliorer l'interopérabilité et la recherche de données dans ce domaine. Ces outils, protocoles, modèles et services de données sont désignés sous l'appellation générique d' « Observatoire Virtuel ». L'observatoire virtuel (OV) est donc une collection d'archives de données interactives (appelées « services OV ») et d'outils logiciels qui utilisent Internet pour bâtir un environnement de recherche scientifique dans lequel les programmes de recherche en astronomie peuvent être conduits. L'OV ouvre de nouvelles voies d'exploration des données astrophysiques issues d'instruments spatiaux, basés au sol, de simulations numériques, etc. Il a pour objectif de gérer de manière transparente l'accès à de grandes quantités de données hétérogènes localisées de par le monde.

[3] https://www.cta-observatory.org/project/status/

[4] http://sci.esa.int/euclid/45403-mission-status/

[5] http://www.ivoa.net/about/what-is-vo.html

Le nombre de services OV est important, ainsi le protocole OV le plus utilisé (ConeSearch) sert plus de dix mille services à lui seul.
Bien que l'intérêt de l'OV ne soit plus à démontrer, son utilisation souffre encore de problèmes :

- de localisation et d'exploitation des services répondant à des critères précis
- de sélection des outils logiciels adaptés aux traitements des données
- d'interopérabilité dus à la pluralité des implémentations possibles pour un même modèle de données, rendant complexe leur comparaison et l'exploitation convenable des informations décrites
- d'automatisation et d'élaboration de chaînes de traitements en fonction des résultats recherchés et des services disponibles
- de traçabilité des traitements opérés sur les données

Des automatisations pour la recherche de services OV et la récupération d'informations ont été mises en œuvre dans les développements liés à l'OV. Toutefois ces dernières ne concernent que des processus déjà connus, pour lesquels les services OV fournisseurs de données et les outils de traitement de ces données sont déjà définis. Elles reposent sur une expertise humaine que la plupart des utilisateurs ne peuvent pas apporter. C'est donc un enjeu majeur de disposer d'une automatisation complète exploitant toute l'étendue des possibilités que propose l'OV indépendamment de cette expertise humaine. Une telle automatisation nécessite une composition de services basée sur une analyse sémantique des capacités exprimées dans les services considérés. Cette analyse sémantique porte sur le sens des informations ou des traitements proposés par un service et sur les liens que cette signification entretient avec les autres connaissances du domaine. Elle devrait permettre à un algorithme de se substituer à l'expertise apportée par l'homme dans le fonctionnement actuel.

L'objectif de ce travail de thèse est de proposer une approche visant à faciliter l'utilisation de services astrophysiques répartis de par le monde. Cette approche se basera sur des méthodes et des processus permettant de rechercher et d'interpréter des données dans le cadre large de l'astrophysique, à un niveau d'abstraction plus haut que les architectures d'Observatoires Virtuels existantes. L'application de ces méthodes et de ces processus devra permettre aux utilisateurs de spécifier et de récupérer les observations dont ils ont besoin, indépendamment de leur connaissance à propos des données à leur disposition dans la masse de services utilisables.

2. Contexte scientifique

La composition de services basée sur la sémantique est étudiée dans la recherche en informatique, notamment dans le domaine des « ontologies ». Les ontologies informatiques sont un des éléments de réponse apportés aux défis posés par l'exploitation de grandes masses d'informations. Elles peuvent être définies comme des « spécifications explicites d'une conceptualisation» (Gruber 1993). Elles occupent une place centrale dans le « Web sémantique » (Berneers-Lee et al. 2001), qui est une évolution du Web existant vers une architecture orientée davantage par la prise en compte du sens des informations contenues dans les documents, plutôt que par leur représentation technique. Le Web sémantique est architecturé autour de plusieurs composants :

- Une représentation sémantique des informations contenues dans les documents et les services Web et de la localisation de ces informations
- Un support informatique pour cette représentation sémantique
- Des agents capables d'exploiter ce support informatique afin de l'utiliser comme support de raisonnement

Le Web sémantique s'appuie sur « l'intelligence » de ces agents pour exploiter la représentation de l'information, afin d'automatiser l'utilisation du contenu des documents et des services. Le but de cette évolution du Web est donc l'automatisation de la sélection des sources d'informations et le rapprochement des contenus. Les services Web décrits au moyen d'ontologies sont appelés services Web sémantiques (*Semantic Web Services*, SWS). Cette représentation sémantique décrit les capacités des services, telles que les fonctions qu'ils peuvent assurer ou les données qu'ils peuvent fournir. Elle décrit également les impératifs préalables à leur utilisation, par exemple les informations d'entrée qui leur sont nécessaires pour fonctionner et les conditions générales qui doivent être satisfaites avant leur invocation. Par exemple, un service de facturation sera invocable uniquement si une commande a été dûment enregistrée au préalable (condition générale à satisfaire) et demandera l'adresse de facturation (donnée d'entrée) pour s'exécuter, puis fournira une facture (donnée de sortie, capacité).

La représentation sémantique des services permet d'établir des enchaînements de services (workflows) par le biais de la composition. La composition de services est l'établissement d'une liste de services compatibles entre eux, qui utilisés conjointement fournissent une information, ou remplissent une fonction donnée. La composition comporte des phases de découverte de services (quels sont les services à disposition), de sélection de services (identifier les services adaptés à la situation et compatibles entre eux) et d'orchestration (déterminer l'ordre d'invocation des services permettant d'atteindre l'objectif fixé). Selon les approches, une composition de services peut être manuelle, semi-automatique ou automatique.
Les services susceptibles de participer à la composition de services visée par ce travail de thèse sont hétérogènes. Certains de ces services sont accessibles depuis le Web, d'autres ne le sont pas et les descriptions de leurs capacités varient suivant le vocabulaire utilisé et le niveau de détail décrit. Les problèmes posés par cette hétérogénéité peuvent être améliorés en fédérant les caractéristiques techniques et les descriptions des capacités des services autour d'une ontologie commune. Par conséquent, le problème de composition de services se ramène à un problème de composition de services sémantiques.
Une structure d'ontologie propre à décrire les compétences et les détails techniques d'invocation des services est donc à rechercher. Cette structure devra pouvoir décrire un domaine de connaissances pour lequel peu de représentations ontologiques préexistantes sont disponibles, et utilisant un vocabulaire très spécifique rencontré dans des textes destinés à des experts du domaine. Les incertitudes liées à cette représentation devront pouvoir être quantifiées, et les éléments de cette représentation devront être correctement associés aux capacités des services. Cette structure et son contenu devront pouvoir servir de support à une composition de services efficace.
La composition automatique de services sémantiques que nous visons sera appliquée au contexte applicatif astrophysique. L'expression des buts recherchés par chaque cas de composition ne devra exiger qu'un minimum d'efforts de l'utilisateur final, et ne nécessiter aucune connaissance technique

préalable. Les résultats devront être compréhensibles, et les paramètres de choix généraux de services accessibles et facilement modifiables. Obtenir une composition répondant aux critères exposés ci-dessus, dans un contexte présentant des caractéristiques telles que celles rencontrées dans les services astrophysiques soulève plusieurs problèmes pratiques. Ces problèmes pratiques sont liés à des problèmes théoriques, que ce manuscrit a pour ambition de détailler et pour lesquels nous avançons des propositions de solutions. Nous nous appuierons sur le contexte applicatif astrophysique pour illustrer les problèmes soulevés et les solutions proposées.

3. Verrous scientifiques

Pour atteindre l'objectif global de cette thèse défini dans le contexte applicatif, il est nécessaire de faciliter l'identification de services adaptés à une requête précise, et de simplifier l'appel aux services comme l'interprétation des résultats produits. Il est également indispensable d'améliorer l'interopérabilité des services astrophysiques afin de permettre leur composition (l'utilisation conjointe de plusieurs services). La description succincte des contextes applicatif et scientifique laisse penser que les principes sur lesquels repose le Web sémantique peuvent apporter de nouvelles pistes pour améliorer l'interopérabilité des services OV. Les ontologies en elles-mêmes, comme leur application pour la constitution du Web sémantique, sont des sujets de recherche d'actualité pour lesquels des verrous doivent encore être levés.

Les services utiles à la production d'un workflow ne sont pas forcément que des services Web. Ils ne sont pas non plus systématiquement issus d'une infrastructure commune (framework d'entreprise, OV pour l'astrophysique…). Des bases de données et des procédures de traitement décrites de façon hétérogène peuvent être sollicitées. Les descriptions des capacités des services peuvent s'appuyer sur des termes différents pour désigner les mêmes quantités physiques, en fonction des pratiques habituelles des distributeurs de données et du niveau de détail précisé. Il est important de définir une couche d'interopérabilité permettant de réunifier ces différences dans une ontologie commune, ce qui amène à l'apparition d'un premier verrou.

- Comment décrire des ressources issues de standards différents, aux profils hétérogènes (bases de données, librairies de traitement scientifiques, services Web…) d'une façon homogène, simple et efficacement utilisable pour la production de flots de traitement automatisés ?

Pour assurer l'identification des services Web adaptés à une requête précise, la description des aspects techniques de ces services doit s'accompagner d'une description de leur domaine d'application. Cette description du domaine d'application revient à modéliser les connaissances que ce domaine contient. Les sources d'information disponibles pour concevoir une structure de représentation de cette connaissance peuvent être constituées de peu de représentations ontologiques préexistantes, tels que pour des domaines très spécialisés. L'essentiel des informations disponibles se trouvent alors à l'intérieur des services eux-mêmes, à destination des spécialistes du domaine. La description de ces informations n'est souvent pas contextualisée, et utilise des termes très précis et très spécifiques. Cette description adopte un style concis, purement informatif et avec des présupposés importants sur la

compétence du lecteur ; ce qui dégrade considérablement les performances des algorithmes existants d'extraction automatique de la connaissance. Cela nous amène à la définition d'un second verrou.

- Comment obtenir une représentation fiable de la connaissance à partir de textes courts, non grammaticalement structurés, liés à un contexte très spécialisé ?

Les propositions qui seront faites pour répondre au verrou précédent imposeront la définition d'une taxonomie possible, dont la qualité demandera à être évaluée. Les concepts de la taxonomie résultante serviront à identifier les capacités des services, et un mécanisme reliant ces capacités à la taxonomie devra donc être proposé. Il est important d'opérer des choix de services qui prennent en compte la confiance accordée aux informations qu'ils fournissent. Cette confiance dépend à la fois de la confiance accordée à la taxonomie elle-même et de celle accordée à la pertinence du rattachement d'un service à un élément de cette structure. En conséquence, les verrous suivants apparaissent :

- Comment quantifier la validité des concepts issus du traitement de ces morceaux de textes ?
- Comment rapprocher ces concepts très spécifiques à un domaine de connaissances d'une description de services sélectionnables et interrogeables?

Les réponses aux verrous précédents amèneront à la définition d'une structure d'ontologie et à une méthode d'enrichissement et de population qui seront argumentées dans ce manuscrit. La composition automatique des services sémantiques contenus dans cette ontologie devra donc tenir compte de cette structure ; or les algorithmes de composition de services existants conviennent mal aux spécificités du type de profils de services que nous avons à traiter.

De plus, l'architecture de services préexistante (l'OV, dans le cas astrophysique) impose un fonctionnement aux services qu'elle contient, notamment l'utilisation de formats et de protocoles de communication définis qui s'écartent des standards habituels rencontrés dans les sciences de l'information. Ce qui amène à la définition d'un dernier verrou.

- Comment parcourir la structure obtenue pour retrouver et sélectionner les services adaptés à une recherche d'information précise?

4. Contributions

En conséquence, les contributions de ce travail de thèse pour lever les verrous susmentionnés s'articulent autour de plusieurs thèmes :

1. **La proposition d'une ontologie de services incorporant un module thématique de description de la connaissance du domaine d'application, et un module spécifique de description des services**. Cette ontologie de services pourra être réutilisée dans d'autres domaines d'application en suivant la même méthodologie que celle décrite dans ce manuscrit.

2. **La définition d'une méthodologie d'enrichissement et de population automatique du**

module thématique créé. L'enrichissement et la population devront se faire à partir de fragments de textes non grammaticalement structurés et au vocabulaire très spécifique. La méthodologie proposée pour cet enrichissement et cette population se satisfait de peu de connaissances ontologiques préexistantes. Elle assure également une quantification de la qualité de la structure obtenue et du peuplement opéré.

3. **La proposition d'un algorithme de composition et d'orchestration de services basé sur le parcours de l'ontologie obtenue par les points précédents**. Cet algorithme devra composer automatiquement des enchaînements d'appels aux services fournisseurs de données et de traitements de ces données. Il devra en conséquence être capable de sélectionner, d'interroger et de faire collaborer des services hétérogènes pour répondre au problème posé. La mémoire des flots de traitement ainsi obtenus et de leur performance estimée par les utilisateurs sera également conservée et réutilisée.

4. **L'élaboration d'une plateforme applicative**. Cette plateforme mettra en œuvre les principes et la méthodologie développés précédemment. Elle proposera une composition automatique de services sémantiques, destinée à des utilisateurs sans connaissance techniques concernant les services utilisés. Elle devra faciliter et apporter un gain de performances pour la recherche d'information dans les services disponibles.

Les solutions théoriques, informatiques proposées dans ce manuscrit seront donc utilisées pour apporter des éléments de réponse concrets aux impératifs du contexte applicatif astrophysique.

5. Organisation du manuscrit

Chapitre 1 : Etat de l'art.

Nous ferons dans ce chapitre un tour d'horizon de l'état des recherches actuelles sur les ontologies, la représentation et l'utilisation de la représentation sémantique des connaissances dans les sciences de l'information. Nous commencerons à exposer les ontologies de services disponibles et les approches permettant la composition de services à partir de services Web sémantiques.
Nous établirons un panorama des méthodes et technologies actuelles permettant l'interopérabilité des données en astrophysique. L'architecture de l'OV y sera détaillée, à travers les initiatives issues de différentes sous-disciplines de l'astrophysique (héliophysique, planétologie…) et les grandes catégories de solutions disponibles. Nous explorerons l'utilisation actuelle des ontologies issues des sciences de l'information à l'intérieur de ces architectures.

Chapitre 2 : Un module ontologique générique pour les services.

Ce chapitre traitera des ontologies de services actuellement disponibles et des spécificités des services que nous avons à représenter. Il présentera les implications de ces spécificités sur la description de services proposée et examinera le module de services de l'ontologie, son rôle et son architecture.

Chapitre 3 : Construction semi-automatique d'un module de domaine.

Ce chapitre sera consacré aux problèmes rencontrés lors de l'enrichissement automatique d'une ontologie à partir de textes courts, très spécialisés et non structurés. Nous verrons les limites des approches existantes lorsque les corpus de textes sont composés suivant cette géométrie, et que peu de ressources externes sont disponibles pour apporter un soutien à l'identification de concepts et de relations. Une méthodologie pour parvenir à peupler la structure enrichie pour obtenir une ontologie utilisable en tenant compte de ces contraintes sera proposée.

Chapitre 4 : Composition sémantique et automatique de services.

Nous étudierons comment l'ontologie définie et peuplée peut être parcourue, pour obtenir une composition automatique de services apte à produire les informations recherchées par les utilisateurs. Nous exprimerons des propositions pour obtenir une composition de services simple d'utilisation, aux critères de composition modifiables et assurant une qualité de service la plus forte possible compte tenu des spécificités du domaine.

Chapitre 5 : Application.

Ce chapitre décrira la plateforme développée sur la base des propositions précédentes, qui représente le résultat utilisable des travaux exposés dans le reste du manuscrit. Les différences entre l'outil proposé et les outils existants seront examinées. Cela permettra de mesurer les apports et les limites de cet outil, pour dégager de futurs travaux propres à améliorer l'inclusion de services partageant des contraintes voisines de celles rencontrées en astrophysique dans le futur Web sémantique.

En conclusion générale, nous ferons un résumé des problématiques scientifiques auxquelles s'est attaché ce travail de thèse, les réponses ou les éléments de réponse qu'il a tenté d'apporter, le gain obtenu et les travaux à envisager pour donner une suite à ces propositions. Les limites et les futurs travaux à apporter à la plateforme proposée aux utilisateurs seront également examinés.

Chapitre 1: Contexte et problématique

1.1. Introduction

Dans ce chapitre, nous allons donner des éléments de contexte permettant d'aborder la suite du manuscrit. Nous nous intéresserons en premier lieu à la définition des ontologies dans les sciences de l'information et au rôle qu'elles jouent dans le Web sémantique, envisagé comme une évolution aussi bien que comme une extension de l'architecture Web actuelle. Cette évolution vise à exprimer les connaissances disponibles dans les services Web non seulement par les données qu'ils contiennent, mais en incorporant également des règles de raisonnement sur ces données. Nous verrons quelles sont les méthodologies existantes pour la construction et le peuplement des ontologies, et nous nous intéresserons au cas particulier des ontologies de services. Le rôle et la structure des principales ontologies de services seront abordés.

Nous verrons ensuite les mécanismes développés pour assurer l'interopérabilité des données et des services dans l'astrophysique. Le concept d'Observatoire Virtuel et certaines de ses principales mises en œuvre seront exposés. Nous verrons quels sont les mécanismes que propose l'OV défini par l'IVOA pour l'interopérabilité des données, la description et la découverte de services. Puis nous donnerons des exemples de logiciels capables d'utiliser les possibilités offertes par l'OV pour faciliter son utilisation par les scientifiques. Enfin, nous examinerons les précédentes investigations de l'utilisation d'ontologies dans le cadre astrophysique.

Ce double état de l'art portant sur le web sémantique et les ontologies d'une part, et sur l'OV et les ontologies en astrophysique d'autre part nous permettra de positionner le travail de cette thèse dans son contexte scientifique comme dans son contexte d'application.

1.2. Web sémantique et raisonnement à base d'ontologies

Les ontologies sont le support pour la représentation sémantique des connaissances soutenant le Web sémantique. La définition du concept d'ontologie en informatique n'est pas récente. Thomas Gruber en donne un énoncé venant de la philosophie, « An *ontology* is an explicit specification of a conceptualization » (Gruber 1993) souvent cité à raison comme « An *ontology* is an explicit specification of a shared conceptualization », tant les aspects portant sur le partage de la conceptualisation et de l'ontologie résultante sont importants dans cet article. Non seulement la conceptualisation doit être partagée, mais elle doit également s'exprimer de façon non ambigüe (en cherchant à éviter toute confusion concernant les termes et concepts employés) et explicite (le plus complètement et clairement possible). L'apparition généralisée des concepts de l'Internet sémantique

au sein de l'Internet existant reste à opérer (avril 2017). Au sein du W3C, le groupe « data activity »[6] est chargé de proposer des pistes de travail (sous la forme de formats d'échange de données, de modèles…) pour permettre de réaliser cette évolution vers une nouvelle forme de Web.

Les ontologies permettent de représenter la connaissance et de raisonner sur la connaissance qu'elles modélisent. Elles sont donc un support important pour tous les domaines scientifiques où la collaboration entre logiciels, personnes ou thématiques est importante. C'est ainsi que plusieurs développements d'ontologies ont été menés dans des domaines scientifiques différents, comme la médecine (Antipolis 2013) ou la biologie (Ashburner et al. 2000). Concevoir une ontologie sans automatisation est une tâche complexe, nécessitant la collaboration de plusieurs acteurs, l'acceptation d'une conception commune par chacun de ces acteurs et une formalisation adéquate. Automatiser toute ou partie de cette conception est donc un objectif scientifique pour le déploiement des ontologies dans un large domaine de champs d'intervention. Les approches visant à apporter cette automatisation se basent sur l'analyse de documents (en particulier, des corpus de textes) et le rapprochement du contenu de la connaissance extraite de ces documents avec des ontologies déjà constituées.

Il existe plusieurs sortes d'ontologies, et les catégories les plus fréquemment reconnues sont les ontologies de haut niveau (upper ontologies), les ontologies de domaine (domain ontologies) et les ontologies hybrides. Les ontologies de haut niveau (par exemple Basic Formal Ontology, BFO (Arp & Smith 2011)) sont des ontologies présentant des concepts communs à plusieurs domaines de connaissances. Les ontologies de domaine visent une représentation d'un champ particulier de connaissances, qui peut être vaste (par exemple, l'archéologie Romaine (Pia 2015)) ou plus spécialisé et ne représenter qu'un sous-ensemble spécifique d'un domaine de connaissances. Il n'existe pas de frontière claire entre les catégories d'ontologies, et on peut considérer qu'une ontologie est « hybride » lorsque la description du domaine de connaissance qu'elle contient peut être réemployée pour le même domaine, dans d'autres contextes que celui dans lequel elle a d'abord émergé. Les ontologies de services ne sont pas fréquemment citées, certainement parce que leur utilisation est moins généralisée que les précédentes, et leur champ de description plus restreint. La distinction entre ontologies de domaine et ontologies de services sera pourtant la plus importante à opérer pour aborder le contenu de ce manuscrit. Les ontologies de services sont une catégorie d'ontologies dont le rôle est de décrire les capacités de services Web (les informations qu'ils peuvent fournir, ou les activités qu'ils peuvent assurer) et leur représentation technique. Le but de cette représentation est d'automatiser l'utilisation des services décrits par ces ontologies, et amène naturellement à la recherche de compositions automatiques de services.

1.2.1. Définition des ontologies en informatique

[6] https://www.w3.org/2013/data/

Une formalisation claire et standardisée des ontologies en informatique est nécessaire. Nous utiliserons la définition suivante, basée sur la formalisation donnée par (Karger et al. 2004) et complétée par (Essaid et al. 2012):

$$O = \{C, H_C, I, R, H_R, R_C, R_I, A, N\}$$

Une ontologie O est définie comme un 8-uplet composé :

- D'un ensemble de concepts C organisés en hiérarchie de classes et sous-classes
- D'un ensemble de relations de subsomption H_C reliant les classes et les sous-classes
- D'un ensemble I des instances des concepts, souvent appelés individus
- De relations logiques binaires R
- D'instances de ces relations entre des concepts, R_C
- D'instances de ces relations entre des instances des concepts, R_I
- Un ensemble de relations de subsomption entre les relations elles-mêmes H_R
- Un ensemble d'axiomes A permettant d'inférer de nouvelles connaissances
- D'un ensemble d'annotations N enrichissant la description des concepts, des relations ou de leurs instances.

La Figure 1 expose une partie de l'ontologie proposée dans ce manuscrit, illustrant ce qui précède. Cette ontologie décrit la connaissance sur le thème de l'astrophysique. Le concept représenté sur la Figure 1 est « absolute band magnitude ».

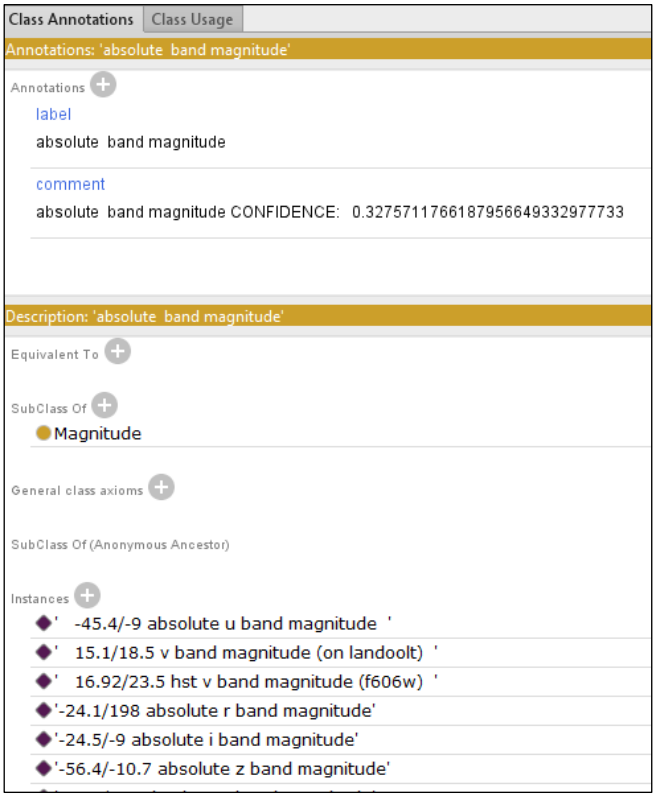


Figure 1: Classe, sous-classe et instances d'une ontologie

Dans une ontologie, concepts, relations et instances peuvent être annotées. Les annotations sont des éléments de texte informatifs explicitant la signification et l'utilité de l'élément annoté. Le concept « absolute band magnitude » est accompagné de deux annotations de l'ensemble *N*, de type « label » et « comment ». La classe « Magnitude » est la classe parent de « absolute band magnitude ». Les instances (souvent appelées « individus ») « -45.4/-9 absolute u band magnitude », « 15.1/18.5 v band magnitude (on landolt) » etc. sont des instances de la classe « absolute band magnitude ».
La Figure 2 représente des relations dans une ontologie. « Thing » est le concept racine de l'ontologie. Les relations sont indiquées sur la partie droite de la figure. Seul le concept « Aggregate » dérive directement de Thing (relation « has subclass » de l'ensemble H_C). Les instances des concepts sont liées par des relations de l'ensemble R_I matérialisées par les arcs entre les concepts. Ainsi, les individus de la classe Aggregate sont combinés à des paramètres de la classe Measurements (relation de R_I IsCombinedToParam, qui a pour domaine ou origine Aggregate et pour Range ou portée Measurements).

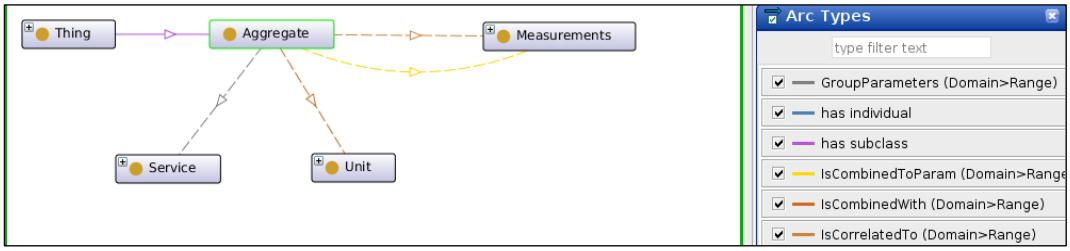


Figure 2 : Relations entre concepts dans une ontologie

Une ontologie est destinée par nature à partager la connaissance qu'elle décrit. Il est donc nécessaire que la conceptualisation et la formalisation de cette connaissance puissent être correctement interprétées, par des agents automatiques ou par des utilisateurs humains. La production d'annotations aussi complètes et aussi lisibles que possible par un utilisateur humain fait partie des facteurs améliorant la réutilisabilité d'une ontologie.

La structure d'une ontologie est un élément qui a un rôle important dans la réutilisabilité, manuelle ou automatique. Suivant la définition donnée précédemment, la structure d'une ontologie s'organise en classes et en sous-classes. Une structure profonde (comportant un grand nombre de sous-classes pour un concept-racine, entre ce concept le plus général et la sous-classe la plus particulière) est de nature à décrire précisément le domaine si les sous-classes sont pertinentes, mais la lecture d'un schéma très détaillé peut devenir complexe. Parcourir la hiérarchie des classes et identifier le niveau de détail approprié pour l'identification d'un concept est ainsi plus difficile dans un schéma très précis et peut nuire à la réutilisation de l'ontologie par des agents automatiques.
La multiplicité des sources de connaissance et des formats, le choix du niveau de détail approprié, les types de raisonnement envisagés et les procédures de maintenance rendent la construction, l'enrichissement et l'évaluation d'une ontologie complexe.

1.2.2. Méthodologies pour la conception d'ontologies

L'utilisation des ontologies dans les sciences de l'information soulève la question d'une méthodologie globale pour leur conception, leur enrichissement et leur évaluation. La construction d'une ontologie est une tâche qui demande des connaissances d'experts du domaine de connaissance à décrire, aussi bien que des compétences d'ingénierie des ontologies elles-mêmes. Il s'agit donc d'une tâche multidisciplinaire, au cours de laquelle la collaboration et la communication entre tous les acteurs impliqués est importante. En réponse à la complexité de cette tâche, différentes méthodologies de conception d'ontologies sont rapidement apparues (López 1999).

Methontology (Baccigalupo & Plaza 2007) s'intéresse par exemple dès la conception de l'ontologie aux cas d'utilisation auxquels elle devra répondre et à son périmètre (qui influence le niveau de détail utile pour les concepts). Cette méthode identifie des phases qui structurent le développement d'une ontologie, de sa spécification à sa maintenance. Ces phases impliquent des activités, pour lesquelles des techniques sont proposées, comme l'analyse de texte ou l'interview d'experts pour l'acquisition de connaissances. Les résultats de ces activités sont notamment présentés dans des documents tels qu'un modèle conceptuel ou un document d'intégration.

D'autres méthodologies comme Neon Methodology (Haase et al. 2008) ont une orientation plus centrée sur les spécificités intrinsèques au développement des ontologies. Le projet Neon comporte la méthodologie accompagnée d'un logiciel (Neon Toolkit, basé sur Eclipse) permettant de la mettre en œuvre. Cette méthodologie propose de prendre en compte l'identification des points communs de l'ontologie avec d'autres ontologies disponibles (appelé « l'alignement » d'ontologies). Une évaluation de l'ontologie est proposée, et le logiciel fournit une interface permettant à un utilisateur d'interagir avec l'ontologie produite.

La méthode « *Grounding Ontologies with Social Processes and Natural Language* » (GOSPL) (Debruyne & Meersman 2012) propose un processus itératif axé sur la collaboration entre les membres de la communauté impliqués dans le développement de l'ontologie. L'ontologie à produire est découpée en sous-ensembles et le processus comporte des phases, dont la succession se répète pour chaque sous-ensemble jusqu'à l'obtention d'une ontologie finale. GOSPL identifie dès le démarrage de la conception des composants externes avec lesquels l'ontologie est destinée à s'interfacer, ou bien qui servent de références pour des choix de vocabulaire. Une grande partie des discussions entre les membres participant au développement consiste à s'assurer que les interfaces prévues soient bien définies. Cette méthode est accompagnée d'un logiciel permettant de proposer des modifications au schéma de l'ontologie, soutenu par un forum de discussion et un système de vote portant sur les modifications suggérées.

Upon Lite (Nicola et al. 2016) partage les ambitions de GOSPL en visant à maximiser les rôle pris dans le développement par les utilisateurs finaux et les experts du domaine. La méthode n'est pas soutenue par un outils spécifique, mais suggère l'utilisation d'outils tiers et bien connus (Goggle Docs, l'éditeur d'ontologies Protégé…) pour la gestion des documents et des discussions. L'objectif principal de Upon Lite est de parvenir à une ontologie dans des délais plus courts que par l'utilisation d'une autre méthode, et en minimisant le rôle des experts de la formalisation des ontologies par rapport à celui des experts du domaine.

Une synthèse de ces différentes méthodes montre que le cycle de vie pour le développement d'une ontologie est défini la plupart du temps en suivant une logique comparable à celle du développement logiciel. Ce cycle de vie s'articule autour d'étapes communes à la plupart des méthodologies (annotation et documentation, acquisition de la connaissance…) auxquelles chaque méthode vient ajouter ses propres impératifs.
L'automatisation de la conception, de l'enrichissement et de la population d'ontologies est également au cœur de recherches récentes, qui seront abordées dans le chapitre 3 de ce manuscrit, « Enrichissement et population automatique de l'ontologie ».

1.2.3. Formats et langages des ontologies

La formalisation de la conceptualisation d'une ontologie est rendue utilisable par une expression technique, compréhensible par des agents informatiques en vue d'une consultation automatique de la connaissance exprimée. Cette expression technique peut être réalisée à l'aide de plusieurs langages. Le choix du langage adopté pour une ontologie dépend notamment du contexte d'utilisation prévu (l'ontologie est-elle destinée à être partagée par le Web, ou utilisée dans un cadre local…). Ce choix est aussi dépendant d'un compromis entre l'expressivité du langage (ce qu'il est capable de décrire), les applications envisagées de l'ontologie et sa lisibilité (par un utilisateur humain).

Une ontologie peut être exprimée dans des langages de type « Knowledge Interchange Format » (KIF)[7]. KIF a pour but de permettre l'échange de connaissances entre des ordinateurs différents, et la syntaxe KIF demande à être traduite dans un représentation interne propre à chacun des ordinateurs pour pouvoir être exploitée. KIF peut également servir de représentation interne, mais cela n'est pas son but principal. KIF n'est pas non plus destiné à fournir une représentation compréhensible par un humain. Il s'agit davantage d'une représentation de la connaissance lisible par un programmeur. Parmi les langages dérivés de KIF, nous pouvons mentionner PowerLoom[8] dont il sera question dans le chapitre 4 : Composition de services.

Lorsque les ontologies sont destinées à une utilisation partagée à travers le Web, il existe des langages spécifiques pour les exprimer. Resource Description Framework (RDF)[9] est un langage permettant d'exprimer des métadonnées à propos de ressources Web (des documents, des services…). RDF permet aussi d'exprimer des éléments qui participent au fonctionnement des services Web, même si ces éléments ne sont pas directement accessibles par le Web (un numéro de carte bancaire pour un site marchand, par exemple). Des concepts variés peuvent donc être décrits en utilisant RDF, qui offre

[7] http://logic.stanford.edu/kif/dpans.html

[8] http://www.isi.edu/isd/LOOM/PowerLoom/documentation/documentation.html

[9] https://www.w3.org/RDF/

également la possibilité de décrire des liens entre ces objets sous la forme de triplets *sujet (concept) – prédicat (relation) – objet (concept)*. RDF Schema est une extension améliorant les capacités de description sémantique de RDF permettant notamment d'exprimer des classes et des sous-classes de concepts. Elle permet également, parmi d'autres améliorations de spécifier quels concepts peuvent être sujets ou objets d'une relation, par les propriétés de « domain » (sujet) et de « range » (objet).

RDF joue également le rôle de syntaxe concrète pour le langage abstrait le plus utilisé pour l'expression des ontologies sur le Web, « *Web Ontology Language* » (OWL) aujourd'hui dans sa version « OWL 2 »[10]. OWL 2 est composé d'une syntaxe fonctionnelle qui spécifie la structure du langage, et accepte plusieurs syntaxes concrètes (OWL/XML, RDF/XML, Manchester et Turtle) qui servent à exprimer les ontologies de manière utilisable. Parmi ces syntaxes concrètes, seule RDF/XML doit obligatoirement être supportée par tous les outils compatibles OWL 2 (comme des éditeurs d'ontologies). La sémantique décrite dans une ontologie OWL2 peut provenir de graphes RDF (on parle alors de OWL 2 full) ou bien utiliser uniquement les capacités de description fournies par OWL2 (on parle alors de OWL 2 DL).

OWL 2 propose trois profils[11], qui sont des compromis entre l'expressivité du langage et les applications envisagées. Chacun de ces profils est un sous-ensemble du langage complet, et le choix du profil à utiliser dépend principalement de la structure de l'ontologie et du type de raisonnement envisagé.
OWL 2 – EL est destiné à des structures comportant une grande part conceptuelle, c'est-à-dire un grand nombre de concepts et de relations, et proportionnellement peu d'instances de ces concepts et de ces relations.
OWL 2 – RL s'adresse à des ontologies dont la structure est à l'inverse de la précédente, et comporte un grand nombre d'instances par rapport au nombre de concepts et de relations définis. En particulier, le profil RL est utile pour raisonner en temps polynomial par rapport au nombre d'instances contenues dans l'ontologie.
OWL 2 – QL permet de raisonner sur une ontologie dont la structure comporte des éléments extraits de schémas de bases de données. Les raisonnements complexes effectués sur l'ontologie sont traduits en requêtes à destination de bases de données, par une correspondance entre le contenu de l'ontologie et la géométrie des bases. Les données elles-mêmes restent donc stockées dans les bases de données, et l'ontologie sert de méta-modèle pour les raisonnements.

[10] https://www.w3.org/TR/2012/REC-owl2-overview-20121211/

[11] https://www.w3.org/TR/owl2-profiles/

La visualisation, la modification et la population d'ontologies par des humains peut être complexe en raison de la verbosité des langages utilisés (RDF, OWL), ou de leur syntaxe peu lisible (KIF). En conséquence, des éditeurs d'ontologies comme Protégé[12] ont été créés pour simplifier ces tâches.

1.2.4. Peuplement des ontologies

Le peuplement des ontologies consiste à emmagasiner des connaissances dans la structure définie lors de la conception. Suivant la définition formelle des ontologies évoquée plus haut, il s'agit d'ajouter des instances (*I*) dans la hiérarchie des concepts (*C*), d'ajouter des instances des relations R_C et R_I, comme de définir de nouvelles relations de l'ensemble *R*. C'est une tâche complexe, au cours de laquelle il est nécessaire non seulement d'extraire de l'information de sources hétérogènes, mais aussi de mettre ces informations extraites en rapport avec la structure et le contenu existant de l'ontologie.

Les approches scientifiques visant à permettre le peuplement automatique ou semi-automatique d'ontologies utilisent plusieurs techniques (Faria et al. 2013), telles que l'analyse du langage naturel (ou *Natural Language Processing*, NLP), l'apprentissage automatique (*Machine Learning*, ML) et l'extraction automatique d'informations *(Information Extraction*, IE). Ces techniques utilisent des sources (un corpus) de connaissances dont on cherche à extraire l'information, accompagnées de sources d'informations structurées externes au corpus et se rapportant au même domaine de connaissances que ce dernier. Ces sources externes de connaissances sont le plus souvent des ontologies préexistantes à l'extraction d'informations visée couvrant un large spectre de domaines comme DBpedia(Bizer et al. 2007). Des bases de données lexicales comme WordNet (Miller 1995) peuvent aussi être utilisées. Les éléments de vocabulaire spécifiques au domaine de connaissances du corpus sont identifiés par l'utilisation de ces sources de connaissance structurées externes au corpus à analyser. Les relations les plus fiables sont sélectionnées parmi toutes les relations candidates en utilisant des calculs de statistiques et de probabilités (Lee et al. 2013). L'extraction d'informations se fait le plus souvent à partir de textes grammaticalement bien formés.

Les méthodes existantes s'adaptent difficilement aux corpus de textes non grammaticalement structurés, dans lesquels les concepts et les relations sont plus difficiles à détecter que dans des corpus grammaticalement cohérents. Deux facteurs viennent augmenter cette difficulté : Le premier est la présence dans le corpus d'un grand nombre de termes très spécifiques à un domaine de connaissances particulier, et le second est le manque de représentations structurées disponibles se rapportant au domaine de connaissance du corpus. Le chapitre 3 de ce manuscrit, « Enrichissement et population automatique de l'ontologie » traitera de ces problèmes.

[12] http://protege.stanford.edu/

1.2.5. Les ontologies de services

Le partage de descriptions de capacités de services Web (les données qu'ils fournissent, les services qu'ils assurent) et l'utilisation automatisée de ces services est souvent envisagée à travers le système d'annuaires *Universal Description Discovery and Integration* (UDDI[13]) et du langage de description *Web Services Description Langage* (WSDL[14]).

WSDL est un standard permettant de décrire les aspects techniques des services Web. Il permet notamment de spécifier, parmi d'autres paramètres les ports à contacter, les fonctions à appeler et les types de messages qu'elles attendent. UDDI est un système d'annuaire, qui peut être privé (propre à une compagnie, ou à un ensemble de services destinés à collaborer) ou public. UDDI référence les éléments WDSL des services inscrits dans l'annuaire et permet de retrouver des services en recherchant par exemple des fonctions spécifiques et d'accéder à leurs descriptions WSDL afin de connaître les modalités d'utilisation et les types de retour définis pour les services trouvés.

Les descriptions UDDI comme WSDL sont uniquement techniques et ne portent pas d'information sémantique sur les données entrantes ou sortantes, ni sur le rôle exact des fonctions disponibles au sein des services.

Avec l'apparition du concept de Web sémantique, les ontologies de service sont apparues pour combler ce manque de représentation sémantique (Adala et al. 2011). Destinées principalement à décrire des services Web, elles sont différentes dans leur intention des ontologies de domaine, qui décrivent des connaissances portant sur un domaine particulier (Terre et environnement, neurologie…). Les ontologies de service se focalisent sur la description des capacités des services qui y sont enregistrés. Les ontologies de service les plus matures sont OWL-S (Martin et al. 2007) et WSMO (Shafiq 2007).

Les ontologies de services ont pour but de permettre la sélection automatique, non plus de données et de connaissances directement mais de services permettant de retrouver des données et des fonctions. L'apport de ce type d'ontologies par rapport au couple WSDL/UDDI concerne la représentation sémantique des connaissances propres aux services. La recherche de services via des ontologies a pour but de permettre la recherche de données et de fonctions par rapport à leur expression sémantique, indépendamment de leur description technique. Ce champ de recherche reste aujourd'hui en grande partie théorique et les cas pratiques d'utilisation sont peu nombreux (Tosi & Morasca 2015) .

La découverte, la sélection et la composition de services par l'intermédiaire d'ontologies fait en conséquence l'objet de nombreux travaux (Bansal et al. 2014; Puttonen et al. 2013; Rodriguez-Mier et al. 2016). L'enjeu de ces recherches est de parvenir à faire interagir plusieurs services en s'assurant

[13] http://uddi.xml.org/resources

[14] https://www.w3.org/TR/wsdl20-primer/

que les données nécessaires à l'utilisation de chacun soient collectées, et que les données collectées par l'ensemble répondent au problème posé par l'utilisateur. De plus, il est nécessaire de sélectionner les services concurrents (accomplissant les mêmes tâches ou fournissant les mêmes informations) en se basant sur leurs critères de qualité propres et les exigences du problème posé.

A) OWL-S

Web Ontology Language for Services (OWL-S) [15] est une spécification définissant une ontologie pour les descriptions de services Web. Cette description est divisée en trois sous-ontologies, chacune se concentrant sur un aspect spécifique. Les aspects décrits, et illustrés dans la Figure 3 sont les suivants:

- Le rôle d'un service (« *Service Profile* », description générique)
- L'accès au service (« *Grounding* »)
- Le fonctionnement du service (« *Service Process* », ou «*Service Model* », modèle de service).

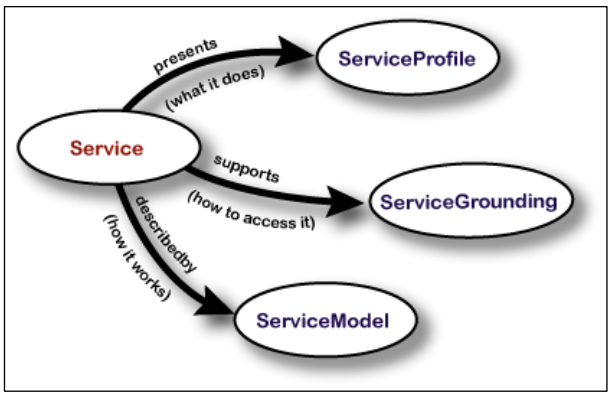


Figure 3: Aspects principaux d'un service dans OWL-S

Les services sont modélisés comme des processus et divisés en trois catégories:

- Des processus simples, qui décrivent des services simples correspondant à une seule requête suivie d'une réponse
- Les processus complexes, qui décrivent les services impliquant un dialogue comportant plusieurs messages échangés entre le client et le service
- Les processus composites qui représentent une composition des services

OWL-S s'appuie sur WSDL (Web Services Definition Language) comme langage de définition des services, comme illustré sur la Figure 4.

[15] https://www.w3.org/Submission/OWL-S/

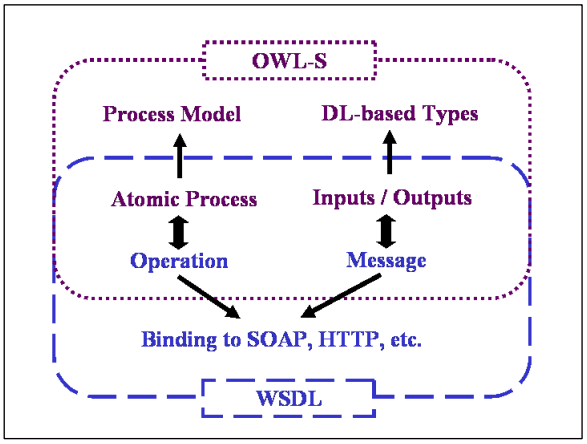


Figure 4: Architecture d'OWL-S et relations avec WSDL

Il est important de noter que OWL-S sépare les sorties des effets et les entrées des conditions préalables. Schématiquement, une sortie (par exemple une mesure) a un effet (par exemple, la mesure est connue) et les entrées (c'est-à-dire les informations nécessaires à un service pour qu'il puisse s'exécuter) sont utilisées par un service seulement lorsque les conditions préalables sont satisfaites. Ainsi, OWL-S décrit les services comme des processus, en utilisant des expressions logiques pour décrire les conditions préalables, les effets et les résultats. OWL-S, en particulier dans les concepts de « *grounding* » d'un service se réfère souvent à l'architecture WSDL / SOAP. Toutefois, les concepts eux-mêmes restent pertinents quelle que soit la mise en œuvre technique et la description des services.

La représentation des services non-WSDL avec OWL-S est difficile (Roman et al. 2015), en raison des multiples références faites à WSDL dans OWL-S. Les services REST notamment, demandent un travail de conversion important avant de pouvoir être exprimés par OWL-S.

B) WSMO

Web Services Modeling Ontology (WSMO [16]) a les mêmes buts que OWL-S, mais s'organise différemment pour y parvenir. La Figure 5 expose les éléments de plus haut niveau de WSMO.

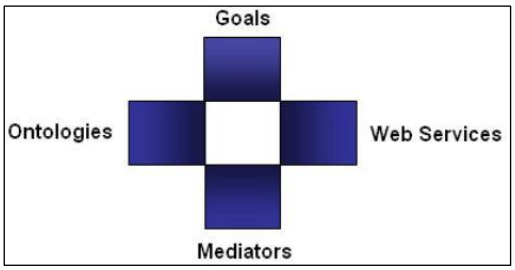


Figure 5: Eléments de haut niveau de WSMO

[16] http://www.w3.org/Submission/WSMO/

Chaque élément de haut niveau dans WSMO décrit un aspect spécifique des services Web. Comme OWL-S, WSMO est une ontologie de haut niveau. Cela signifie qu'une compréhension commune des éléments décrits par chacun des services Web exprimés par ces ontologies doit être assurée. En conséquence, le domaine de connaissances spécifiques aux services Web décrits dans ces ontologies doit être décrit, au cas par cas.

Une vue d'ensemble des éléments de haut niveau composant WSMO est la suivante:

- Les « *ontologies* » décrivent la connaissance du domaine dont les services enregistrés traitent.
- Les « *goals* » sont très clairement définis dans la soumission elle-même: «Les buts sont des représentations d'un objectif pour lequel l'accomplissement est recherché par l'exécution d'un service Web ».
- Les « *mediators* » sont utiles lorsque différentes ontologies décrivent des concepts issus du même domaine de connaissances. Ils assurent les rapprochements basés sur les correspondances entre ces différentes ontologies.
- « *Web Services* » contient les descriptions de base des services Web dans WSMO. En ce qui concerne les buts, nous pouvons nous référer à la description donnée dans la soumission: «les descriptions de services Web dans WSMO se composent de fonctionnalités, de paramètres non-fonctionnels et des aspects comportementaux d'un service Web ».

WSMO se concentre sur la « *choreography* » (le dialogue entre l'utilisateur et le service) comme OWL-S, mais ajoute l'orchestration de manière explicite, en indiquant *«[…]comment le service Web utilise d'autres services Web pour atteindre les fonctionnalités requises»* (Extrait de la documentation).
WSMO est une façon complète, mais plutôt complexe de décrire les services. Son exhaustivité implique une division en plusieurs couches d'ontologies et l'utilisation de multiples expressions logiques à l'intérieur des différents niveaux de descriptions pour assurer leur cohérence. Ces expressions logiques doivent être construites sur un ensemble spécifique de vocabulaire et de termes formant un langage intégré dans WSMO, appelé WSML.
Dans (Roman et al. 2015), les auteurs se concentrent sur une ontologie de services simplifiant WSMO, qui utilise les annotations issues du mécanisme SAWSDL[17] pour faire directement référence aux éléments de l'ontologie dans la description des services eux-mêmes. Ce modèle appelé WSMO-Lite est une proposition du *World Wide Web Consortium* (W3C) depuis 2010 et dérive de WMSO à partir duquel il extrait et réorganise un sous-ensemble afin de supporter les défis du monde réel dans l'intégration des services Web sémantiques. Les buts et les médiateurs ne sont pas présents dans ce modèle qui vise à fournir une ontologie de service légère.

[17] https://www.w3.org/TR/sawsdl/

1.2.6. Composition de services

Les services Web sémantiques (*Semantic Web Services*, SWS) (Parsia 2003) utilisent des langages spécifiquement conçus pour exprimer la sémantique telle que DAML-S (dont OWL-S est une évolution) ou WSMO, et des langages pour exprimer les règles de raisonnement (SWRL)[18] ou effectuer des requêtes comme SPARQL[19] et SQWRL(O'Connor & Das 2009). La composition des services Web sémantiques est un sous-domaine de la composition des services Web. Bien que les recherches récentes sur la composition de services Web aient pu présenter quelques succès réels, la variante sémantique de cette composition de services reste principalement théorique, les SWS n'étant pas encore largement adoptés (Pedrinaci & Domingue 2010) malgré la disponibilité de OWL-S et de WSMO. L'une des raisons expliquant cette lente adoption des SWS est la difficulté rencontrée pour annoter correctement les services Web existants pour les transformer en SWS (Pedrinaci & Domingue 2010), (Tosi & Morasca 2015). Ceci est corroboré par le fait que dans une étude récente sur les plateformes de composition de services Web (Milanovic & Malek 2004), une seule plateforme basée sur SWS a été étudiée parmi les douze explorées dans la vue générale proposée.

Néanmoins, les travaux de recherche concernant l'utilisation de services Web sémantiques proposent des solutions. Ces travaux traitent soit de certains aspects du processus global de composition, soit de l'ensemble des étapes de composition. Les limitations des travaux existants portent sur la difficulté d'exprimer des SWS partageant une représentation commune de leur domaine de compétences (c'est-à-dire, du domaine dans lequel s'inscrivent les fonctions des services ou les informations que les services fournissent). Assurer la meilleure qualité de composition de services possible est une difficulté supplémentaire. Cela revient à prendre en compte les points forts et les points faibles de chaque service en regard des objectifs de la composition, et ces points forts et ces points faibles dépendent de nombreux paramètres (le contexte d'exécution, les retours d'utilisateurs…). Le chapitre 4 de ce manuscrit, « Composition de services » explorera les difficultés rencontrées pour identifier et quantifier ces paramètres.

Le paragraphe suivant présentera les travaux existants permettant l'interopérabilité en astrophysique, et permettra de mettre en perspective ces travaux avec le panorama des travaux sur le web sémantique exposés ci-dessus.

[18] https://www.w3.org/Submission/SWRL/

[19] https://www.w3.org/TR/sparql11-overview/

1.3. L'interopérabilité en astrophysique

L'objectif de ce paragraphe est de présenter succinctement les éléments de compréhension permettant d'appréhender les principales spécificités des services prises en compte vis-à-vis des données et de leur domaine d'application.

1.3.1. Eléments de contexte

L'astrophysique est une science portant sur l'étude des objets présents dans l'univers à travers leurs propriétés et les processus physiques à l'œuvre au sein de ces objets.

L'astrophysique est observationnelle, elle entraîne la création de nombreux procédés de récolte des informations émises par les objets qu'elle étudie. La plupart des informations récoltées sont issues d'émission de rayonnements électromagnétiques produits dans différentes longueurs d'ondes (des rayons gamma aux ondes radio en passant par l'ultraviolet, le rayonnement visible etc.). Les instruments astrophysiques, sont les dispositifs permettant la récolte de ces informations (télescopes au sol et spatiaux, radiotélescopes…). Ils sont le plus souvent composés d'un moyen de captation de cette information (par exemple, un télescope) et d'un instrument d'analyse de l'information captée (par exemple, un spectrographe, un polarimètre, une caméra d'imagerie…). Le terme d'instrument désigne selon le contexte le moyen de récolte (le télescope), le moyen d'analyse (le spectrographe) ou l'ensemble formé par les deux composants.
Beaucoup de télescopes modernes peuvent héberger plusieurs instruments, permettant d'observer dans les mêmes conditions (même site géographique, même moyen de récolte) les objets selon des méthodes différentes (imagerie, spectrographie…).

L'astrophysique est théorique, car elle propose des calculs abstraits visant à modéliser le comportement d'objets ou de phénomènes concrets (des éruptions solaires, des formations d'étoiles…). Ces calculs théoriques servent à vérifier la validité des modèles et des théories comparées aux observations réelles. Ils servent aussi à décrire des phénomènes ou des objets abstraits prédits par les théories (les ondes gravitationnelles, prévues par la théorie de la relativité générale dès 1916 et observées en 2015, le boson de Higgs…) participant à la validation ou la réfutation des dites théories.

Les instruments importants construits pour la production d'observations astrophysiques, notamment ceux pour lesquels la définition et la construction nécessitent plusieurs années par des consortiums internationaux sont le plus souvent accompagnés dès la conception par la définition de centres de données dédiés. Ces centres ont pour rôle de permettre aux scientifiques de bénéficier de procédures de traitement et d'analyse des données avancées, conçues spécifiquement pour le profil des données produites par les instruments concernés. On parle alors de «Science Gateways» (SGN). Les données ainsi traitées (par opposition aux données dites « brutes » qui désignent les données natives produites par l'instrument, avant analyse scientifique) sont mises à disposition du public selon des délais et des procédures définies lors de la conception des instruments. Elles peuvent alors être utilisées par toute la

communauté astrophysique, et non nécessairement par des spécialistes. Dans ce cas, pour qu'elles puissent être utilisées efficacement, elles doivent être décrites si possible de façon compréhensible et standardisée. Ces données publiques sont alors partagées le plus souvent au moyen de services Web. Ces services Web doivent donc, selon la même logique être eux-mêmes documentés et identifiables. Ce sont ces impératifs qui ont conduit à l'apparition du concept d'observatoire virtuel (OV).

1.3.2. L'observatoire virtuel

Devant la croissance toujours plus importante de la quantité de données scientifiques produites par les instruments modernes, l'astrophysique a proposé le concept d'observatoire dit « virtuel » (OV). Le but de cette architecture est de permettre le partage de données scientifiques produites par des instruments issus de multiples domaines des sciences de l'univers, comme l'astrophysique et la planétologie mais aussi l'héliophysique, l'astrochimie... Cette architecture cohabite avec des bases de données spécifiques aux instruments, spécialement dans le cas de « grands » instruments évoqués ci-dessus, ou de grands programmes scientifiques. Ces bases de données spécifiques permettent alors d'opérer des recherches plus précises, d'utiliser des descriptions plus spécifiques ou de bénéficier d'outils de récupération et d'analyse des données dont la description ou l'utilisation n'est pas proposée par l'architecture de l'OV.

Le terme d'Observatoire Virtuel est utilisé aussi bien pour désigner le concept général que chacune des déclinaisons développées pour un champ disciplinaire spécifique. L'organisation et le développement de l'OV diffère effectivement, dans des proportions plus ou moins importantes en fonction du contexte scientifique dans lequel il est mené. Ces développements spécifiques peuvent être des déclinaisons de l'OV le plus abouti (dédié à l'astrophysique) avec des adaptations dues aux impératifs locaux, ou bien des initiatives plus éloignées voire totalement séparées. On peut distinguer plusieurs catégories majeures d' OVs qui partagent souvent le même format (VOTable) mais rarement ou partiellement les mêmes modèles de données et les mêmes protocoles de recherche et de récupération de données :

- L'astrophysique avec l'IVOA (*International Virtual Observatory Alliance*[20]), l'architecture la plus élaborée et dont l'influence est nette sur les autres domaines

- L'héliophysique, la planétologie et les géosciences avec « CASSIS » (*Coordination Action for the integration of Solar System Infrastructures and Science*[21]) regroupant Helio[22], Europlanet RI et SOTERIA

[20] http://www.ivoa.net/deployers/intro_to_vo_concepts.html

[21] http://cassis-vo.eu/activities/index.php

[22] http://www.helio-vo.eu/capabilities/

- L'astrochimie avec VAMDC (*Virtual Atomic and Molecular Data Center*[23]) utilise une version modifiée (VAMDC-TAP) du protocole TAP de l'IVOA mais en est indépendant et possède son propre modèle de données, XSAMS(Dubernet et al. 2016).

- Enfin, concernant la planétologie on peut citer IPDA [24] (*Inter Planetary Data Alliance*) (Sarkissian et al. 2016) qui fait l'équivalent du travail de l'IVOA pour l'astrophysique, mais en planétologie.

Les outils mis à la disposition des communautés scientifiques pour interroger les services contenus dans ces OVs sont propres à chaque grande catégorie, et même à l'intérieur de ces catégories les technologies utilisées sont hétérogènes et sont l'objet de recherches, de tests et de développements ciblés sur des problématiques propres. Il est à noter que ces initiatives ne sont pas ignorantes les unes des autres, et que CASSIS par exemple s'intéresse aussi bien à ce que produit IPDA qu'à ce que produit l'IVOA.

Toutes ces architectures se situent à des niveaux de maturité différents, et sont en proie à des préoccupations liées à leur propre domaine d'expertise, ou bien communes aux autres systèmes. Chacune développe ses propres réflexions et perspectives, même si les échanges sont fréquents entre les disciplines.

La vivacité des recherches de solutions de plus en plus efficaces dans les domaines des OVs est bien illustré par l'exemple du VSTO[25] (*Virtual Solar Terrestrial Observatory*), une initiative américaine bien à part de celles qui précèdent, soutenue par la National Software Foundation (NSF) et à base d'ontologies dont la première version date de 2007 et la dernière de 2011[26].

Le panorama des services Web utiles aux astrophysiciens, toutefois, ne s'arrête pas aux seuls OVs et de nombreux observatoires physiques fournissent des suites d'outils en ligne permettant de traiter, récupérer et comparer des données. Ces données ne peuvent parfois pas être décrites de manière complètement satisfaisante dans les standards et formats actuellement proposés par les différents OVs (on peut rencontrer des problèmes de modèles de données incomplets pour des observations particulières, ou au contraire trop spécifiques pour des modèles théoriques par exemple). Les modèles de données évoluent, mais ne sont pas toujours à même de répondre efficacement aux besoins de descriptions adaptés pour toutes les données à exprimer.

---

[23] http://www.vamdc.eu/documents/standards/dataAccessProtocol/vamdctap.html

[24] http://planetarydata.org/standards

[25] www.vsto.org, http://tw.rpi.edu/Web/project/VSTO

[26] http://tw.rpi.edu/Web/project/VSTO/Releases

Comme évoqué précédemment, il existe une gamme de services proposant conjointement l'utilisation de formats et de protocoles OV et des formats, des outils d'analyse ou de recherche de données spécifiques à certains instruments. C'est le cas de SOHO science archive[27] pour le satellite solaire SOHO, qui propose de récupérer des données au format IVOA (VO Tables), sans passer par un protocole IVOA mais par une interface dédiée, Web ou service Web. NED[28] (*NASA Extragalactic Database*) est un autre exemple qui propose des services OV et non-OV avec des outils en ligne comme un calculateur de correction de vitesses pour les objets astrophysiques[29].

Ces services ne peuvent pas, dans l'état actuel de l'avancement de l'OV être inclus dans les possibilités offertes aux utilisateurs. La raison est qu'aucun mécanisme n'existe aujourd'hui dans les OVs pour décrire des méthodes de traitement de données ou de calculs astrophysiques généraux. Même si plusieurs outils performants « compatibles OV » sont utilisés pour récupérer, parfois traiter et afficher le contenu des données retrouvées dans les OVs, examiner les possibilités offertes en-dehors de l'architecture couverte n'est pas proposé.

1.3.3. L'Observatoire Virtuel selon l'IVOA

L'architecture d'OV la plus aboutie et la plus utilisée est celle définie par l'IVOA. Elle propose des standards de définition de services, des modèles de données et des protocoles d'accès aux services. Ces standards, modèles et protocoles constituent le socle commun sur lequel s'appuie l'interopérabilité des données et des services. La Figure 6 représente l'architecture de l'IVOA telle que présentée dans la recommandation pour l'interface d'accès aux données (DALI) (Dowler et al. 2014).

Le schéma de la Figure 6 est organisé en blocs. Le bloc supérieur comportant les utilisateurs et des ordinateurs (qui représentent une utilisation manuelle ou automatique de l'OV) interagit avec la couche utilisateur « USER LAYER ». Cette couche utilisateur comporte les différents moyens d'accès à l'OV, que ce soit par services Web, par logiciels dédiés ou par scripts. Le bloc inférieur présente les producteurs de données (Providers) qui interagissent avec la couche « RESOURCE LAYER » pour mettre leurs données à disposition.

Les annuaires de services sont représentés sur le bloc gauche du schéma, les protocoles d'accès aux services dans le bloc situé à droite. Le bloc central concerne les éléments soutenant cette architecture, et notamment les modèles de données.

[27] http://ssa.esac.esa.int/ssa/aio/html/howto.shtml

[28] http://ned.ipac.caltech.edu/

[29] http://ned.ipac.caltech.edu/forms/vel_correction.html

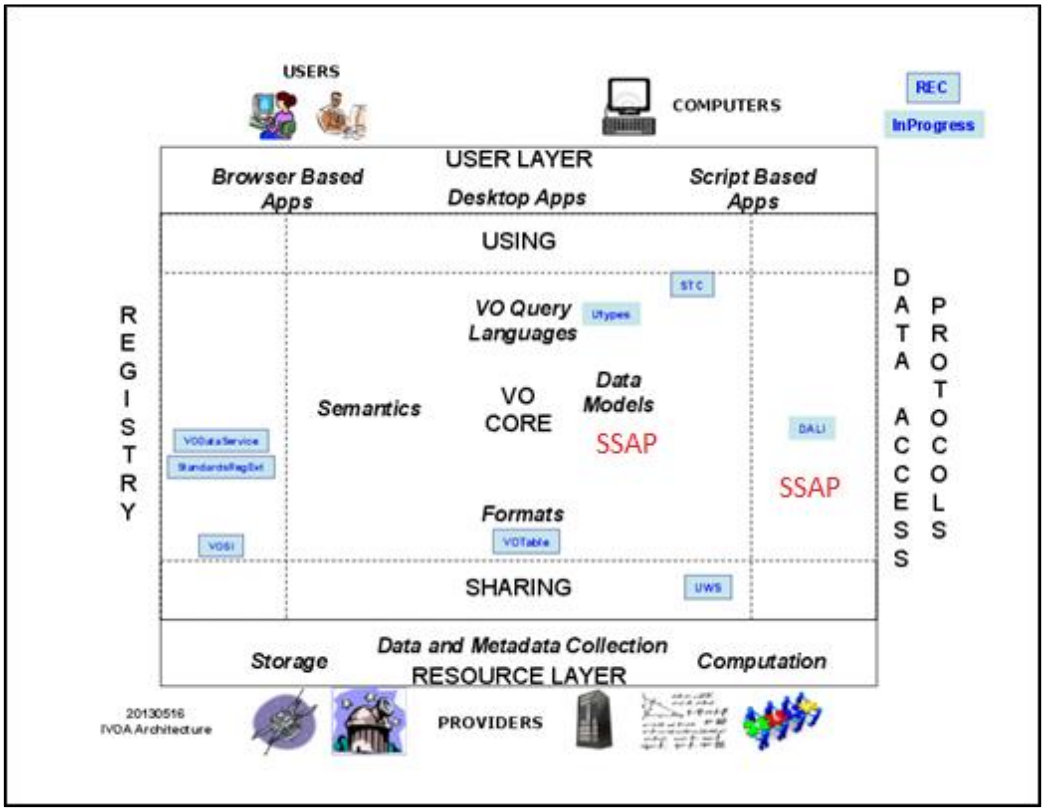


Figure 6: Architecture de l'IVOA

Les modèles de données[30] de l'IVOA sont des schémas permettant la description des métadonnées associées aux données observationnelles ou théoriques. Ces schémas peuvent évoluer au cours du temps en fonction des retours des fournisseurs de services et de leurs cas particuliers d'utilisation. La distinction entre modèle de données et protocole d'accès est parfois floue. Ainsi, Simple Spectrum Access Protocol (SSAP) est à la fois un protocole et un modèle de données.

Deux éléments définis par l'IVOA permettent d'identifier le rôle des éléments d'un modèle et les quantités astrophysiques présentes dans les données : Il s'agit des UTYPEs et des *Unified Content Descriptors* (UCDs).
Les UTYPEs identifient des éléments dans un modèle de données. La forme et la définition des UTYPEs dépendent des modèles de données dans lesquels ils s'expriment. Leur périmètre d'utilisation est celui de la définition du rôle des éléments dans un modèle. Leur rôle est de permettre de rapprocher sur des éléments communs deux documents instanciés depuis le même modèle de données.
Les UCDs désignent des quantités astrophysiques, suivant un vocabulaire fixé par les standards de

[30] http://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaDataModel

l'IVOA[31]. Un extrait de ce vocabulaire est donné dans le Tableau 1. Les UTYPEs peuvent être considérés plus généraux que les UCDs, on trouve par exemple dans SSA les UTYPEs du Tableau 2.

Tableau 1 : Exemples d'UCDs

UCD	Signification
phot.flux	**Photon flux**
phot.flux.bol	**Bolometric flux**
phot.flux.density	**Flux density (per wl/freq/energy interval)**
phot.flux.density.sb	**Flux density surface brightness**
phot.flux.sb	**Flux surface brightness**
meta.bib.bibcode	**Bibcode**

Tableau 2: Exemples d'UTYPEs

UTYPE	*Description*
Char.FluxAxis.Ucd	ucd for flux
Char.SpectralAxis.Ucd	ucd for spectral coord
Target.Name	Target name
Dataset.DataModel	Datamodel name and version
Curation.Reference	URL or Bibcode for documentation

Les deux premiers UTYPEs font référence à des UCDs. Ils indiquent, dans le vocabulaire fixé par l'IVOA pour les UCDs, quels mots sont utilisés pour décrire les quantités de flux et de coordonnées spectrales qui figurent dans les données. Le troisième indique le nom de l'objet cible de l'observation concernée. Dans certains cas, les deux systèmes UCDs et UTYPEs peuvent désigner la même information :

- L'UTYPE Curation.Reference défini dans SSA et l'UCD meta.bib.bibcode désignent tous les deux le Bibcode de la documentation associée aux données.
- Un autre exemple concerne l'UTYPE Target.Redshift et l'UCD src.redshift , qui désignent tous les deux le décalage vers le rouge de la lumière émise par l'objet observé.

Bien que le rôle des UTYPEs et celui des UCDs soient différents, ces redondances peuvent être source de confusions. Les schémas XML proposés par l'IVOA pour la définition de services en fonction des protocoles utilisés (SSA[32], ConeSearch[33] par exemple) sont souples, comme l'illustre le Tableau 3.

[31] http://www.ivoa.net/documents/latest/UCDlist.html

[32] http://www.ivoa.net/xml/SSA/v1.0

[33] http://www.ivoa.net/xml/ConeSearch/v1.0

Tableau 3 : Extrait de définitions de services IVOA

Service 1
<column> <name>Hmag</name> <description>? 2MASS H magnitude (1.6um)</description> <unit>mag</unit> <ucd>phot.mag;em.IR.H</ucd> </column> <column> <name>e_Hmag</name> <description>? Mean error on H magnitude</description> <unit>mag</unit> <ucd>stat.error;phot.mag;em.IR.H</ucd> </column> <column> <name>Kmag</name> <description>? 2MASS Ks magnitude (2.2um)</description> <unit>mag</unit> <ucd>phot.mag;em.IR.K</ucd> </column>
Service 2
<column> <name>Kmag</name> <description>? DENIS Ks-band magnitude (5)</description> <unit>mag</unit> </column> <column> <name>Kcorr</name> <description>DENIS Ks-band correlation factor</description> </column> <column> <name>Kxpos</name> <description>X-position in DENIS K-band image</description> <unit>pix</unit> </column>

La structure du schéma est respectée, mais chaque service implémente le schéma suivant son propre profil. On voit que le service 1 précise l'UCD pour les quantités décrites, là où le service 2 ne le fait pas. Le nom des quantités est décrit suivant des habitudes différentes (Ks-band ou simplement Ks) et des degrés de précision différents (le service 1 précise une longueur d'onde, pas le service 2).

L'OV apporte des réponses aux problèmes soulevés par l'hétérogénéité des données et propose des protocoles d'interrogation communs pour des services fournisseurs de données. Toutefois, des limitations existent toujours :

- La souplesse des schémas de description de services utilisés par l'IVOA vient du nombre limité de mots-clés obligatoires. En contrepartie, cette souplesse génère de l'imprécision et une forme d'hétérogénéité puisque chaque service peut enrichir sa propre description suivant ce que le format lui propose. Des services voisins peuvent donc renseigner des informations communes avec un niveau de détail différent.
- Les modèles de données existants sont parfois insuffisants en ce qui concerne la description de certains types de données particulières. La définition d'un nom de cible obligatoire pour un modèle de données de spectres n'a par exemple pas de sens, pour des spectres théoriques qui ne correspondent pas à l'observation d'une source réelle. Pour certains domaines de recherche (astrophysique rayons gamma), les possibilités permettant de décrire les observations effectuées sont limitées. C'est pourquoi des initiatives comme Helio pour l'héliophysique apparaissent, et aujourd'hui les observations à hautes énergies type gamma sont à un croisement : soit elles choisissent d'intégrer l'IVOA en définissant un modèle de données adapté, soit elles s'organisent différemment mais pour le moment il n'existe pas de « OV hautes énergies ».
- Un autre problème est celui de la connaissance de l'existence des services. Les annuaires actuels fournissent une liste de services répondant à certaines caractéristiques mais l'utilisateur doit faire le tri par ses propres moyens dans cette liste de possibilités qui peut être longue (plusieurs milliers de services).
- Les services implémentant les standards et protocoles de l'IVOA ne sont pas exprimés dans le langage WSDL. La découverte de leurs capacités et l'automatisation de leur utilisation doit donc tenir compte des standards et protocoles internes à l'IVOA, et ne sont pas couvertes par les travaux de recherche en informatique portant sur la composition (automatique ou semi-automatique) de services Web.
- Le cloisonnement des services à l'intérieur de l'IVOA est une autre limite : bien que ces services partagent la même logique globale, rien ne permet par exemple de passer d'une donnée d'imagerie à une donnée spectroscopique ou d'une donnée radio à une donnée infra-rouge de façon automatique. Et dans le cas de deux services proposant des données de même type (des spectres, par exemple) et dans la même longueur d'onde, rien ne permet de les mettre l'un en relation avec l'autre, et un utilisateur accédant à l'un des services ne sera pas informé de l'existence du second.

Le protocole « *DataLink* »[34] constitue la réponse de l'IVOA à la dernière limite évoquée, celle du cloisonnement des services. Ce protocole propose de définir pour un service quels sont les autres services de l'OV qui peuvent être complémentaires. Ce nouveau protocole susceptible d'améliorer la situation du morcellement des services dans l'OV a deux limitations principales :

- Il suppose le développement d'un nouveau service ou l'extension d'un service existant pour fournir ce lien

[34] http://www.ivoa.net/Documents/WD/DataLink-20120419.html

- Les services liés le sont sur la base des connaissances du programmeur du protocole. Bien que cette connaissance garantisse l'adéquation des données entre elles jusqu'à un très haut niveau de détail, ce niveau de détail doit être défini et contrôlé par un expert. De plus, il peut exister d'autres services également utiles qui ne seront pas liés, en raison de la difficulté à prendre en compte le grand nombre de services disponibles (plus de 12000).

A) LES ANNUAIRES

Les « *registries* » sont des annuaires dans lesquels sont enregistrés les services OV. Chaque organisation proposant des services OV peut créer son propre annuaire et on en compte donc plusieurs (l'Euro-VO, NVO et AstroGrid sont parmi les plus connus). Les différents annuaires existants sont parfois limités aux services développés par une seule entité. Parfois, ils sont plus généraux et ils regroupent tous les fournisseurs de services qui les contactent pour s'enregistrer et dont la description technique est validée. Pour permettre aux utilisateurs du monde entier de venir contacter son jeu de services une fois qu'ils sont opérationnels et utilisables, une entité peut donc :

- soit créer son propre annuaire et l'enregistrer dans le « *registry of registries* » (RofR) qui les compile tous
- soit enregistrer directement ses services dans un annuaire plus général (Euro-VO, AstroGrid…), dont le contenu est référencé dans le RofR.

Pour accéder à tous les services OV disponibles, un utilisateur doit donc d'abord connaître l'existence de cet annuaire général et savoir comment l'interroger et interpréter ses résultats. Dans le but de simplifier cette interrogation, VO-Paris a développé un outil permettant d'interroger directement le RofR[35] suivant des mots-clés faisant référence tant à la description technique des services vue dans les annuaires qu'à celui de leur description générale à l'attention des utilisateurs humains.

Cette initiative possède de nombreux avantages, comme celui de proposer un point unique d'entrée aux utilisateurs et de permettre des requêtes de sélection de services plus précises que celle que l'on peut trouver dans un annuaire seul, exprimées suivant une méthode unique. Toutefois le nombre de services répondant à des critères peut être important (plusieurs centaines), et identifier des correspondances ou des complémentarités entre les données de services différents reste à l'entière charge de l'utilisateur, comme l'interrogation de chacun de ces services. Annoter les services en fonction de leurs retours et de la qualité estimée pour chacun des retours en fonction des caractéristiques propres à l'instrument dont les données sont issues est également impossible. Il reste par conséquent nécessaire à l'utilisateur de connaître le fonctionnement de cette application d'interrogation du RofR, et la disparition des détails techniques de l'architecture OV pour l'utilisateur n'est pas encore atteinte.

[35] http://api.voparis-tmp.obspm.fr/registry/

De la même façon que les services de l'IVOA ne sont pas exprimés en WSDL, les annuaires de l'IVOA n'utilisent pas UDDI. Les clients disponibles pour interroger les annuaires UDDI ne sont donc pas utilisables pour trouver ces services particuliers, et cette tâche de découverte de services est laissée à la charge des développeurs de logiciels compatibles OV. Le protocole RegTAP[36], destiné à offrir une interface structurée pour les requêtes aux annuaires de l'IVOA a été finalisé récemment (fin 2014) et son adoption dépendra de la réactivité des développeurs de logiciels.

B) LES APPLICATIONS OV ET LES ANNUAIRES

La diffusion des possibilités offertes par l'OV passe par la production d'outils logiciels abordés au paragraphe précédent. Promouvoir ces outils au sein de la communauté astrophysique nécessite de les décrire, de cerner leur périmètre d'activités et ce qu'ils peuvent apporter de fonctionnalités nouvelles. L'idée d'un annuaire des applications qui jouerait le même rôle que les annuaires de services commence simplement à se développer, et n'est matérialisée (Février 2017) qu'au travers d'un site Internet proposant une liste de logiciels compatibles OV [37] implémentant le protocole Simple Application Messaging Protocol (SAMP). Ce protocole SAMP permet d'échanger des données et des métadonnées entre applications. Il est spécifique à l'IVOA.

Plusieurs questions sont soulevées par l'émergence d'un annuaire des applications. Doit-il calquer son fonctionnement sur celui des annuaires de services ? Doit-il inclure des applications n'implémentant pas le protocole SAMP mais capables d'interroger les services de l'OV ? Doit-il proposer des applications non-compatibles OV mais utilisables sur des données astrophysiques particulières, et sur quelle base de descriptions ? Comment son fonctionnement devrait-il être intégré aux outils eux-mêmes ?

1.3.4. Les logiciels compatibles OV

Parallèlement aux services de données et à l'offre de protocoles et de formats d'interrogations et de descriptions disponibles, les organismes de recherche liés au développement des OVs comme le Centre de Données de Strasbourg (CDS) proposent des logiciels adaptés à cette infrastructure. Ces logiciels permettent de faciliter l'utilisation des OVs, et par conséquence d'augmenter la diffusion et la valorisation des données proposées par les services inscrits. Nous avons vu que l'OV est une architecture proposant des standards, des protocoles et des modèles. Les logiciels compatibles OV sont des outils permettant la manipulation de ces standards et de ces modèles, comme l'utilisation de ces

[36] http://www.ivoa.net/documents/RegTAP/20141208/REC-RegTAP-1.0.html

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

protocoles. Ces outils peuvent être très spécifiques (CASSIS[38], VOSpec[39] pour la visualisation et l'analyse de spectres, TOPCAT[40] pour l'utilisation de catalogues et de tables…) ou bien plus généraux (Aladin[41], conçu à la fois comme un atlas du ciel et un portail vers l'OV).

Le logiciel «*Virtual Observatory oriented SPECtrum workFLOW*» (VOSPECFLOW) (Lèbre et al. 2012) donne un exemple d'application utilisant les formats et protocoles de l'OV pour un cas scientifique prédéfini, embarqué dans un logiciel spécifique. Les bases de données utilisées dans VOSPECFLOW sont interrogées par le biais de protocoles OV et les données récupérées dans un format OV. Toutefois, les spécificités des spectres utilisés dans ce logiciel ont requis un développement nouveau, car aucun logiciel compatible OV préexistant ne convenait au besoin.

Certains logiciels compatibles OV permettent de localiser des services et de les interroger à partir de leurs protocoles, mais les méthodes tri parmi tous les fournisseurs de données disponibles restent à un niveau d'abstraction très général. Cette sélection des services est automatisée jusqu'à un niveau de détail variable suivant les logiciels concernés. Les systèmes utilisés reposent sur l'utilisation de mots-clés recherchés dans les descriptions de services, ou bien sur le type de données retournées par les services (spectres, images, ou catalogues sont les plus utilisés).

Un utilisateur peut donc aujourd'hui rechercher des services dans l'OV, que ce soit au travers des interfaces existantes des annuaires ou par l'utilisation de logiciels existants comme Aladin(Boch & Fernique 2011) ou TopCat (Taylor 2005). AstroTaverna est un cas particulier d'application intégrant des services OV. Il s'agit d'un plugin pour le gestionnaire de workflows Taverna, dont il sera question dans le chapitre 5. Nous allons succinctement présenter ici Topcat et Aladin, qui sont deux logiciels plus utilisés que ne l'est AstroTaverna.

A) TOPCAT

TopCat est l'outil le plus évolué concernant les facilités de recherche de services. Là où Aladin n'offre comme critère que le type de données recherché (spectres, catalogues ou images) TopCat propose une recherche sur la base de mots-clés, combinés ou non. La figure 7 illustre le résultat d'une recherche de services par TopCat, où l'on ne voit apparaître qu'un seul service pour le mot clé « heliocentric radial velocity » alors que bien d'autres services sont susceptibles de fournir cette information.

[38] http://cassis.irap.omp.eu

[39] http://esavo.esac.esa.int/Webstart/VOSpec.jnlp

[40] http://www.starlink.ac.uk/topcat/

[41] http://aladin.u-strasbg.fr/

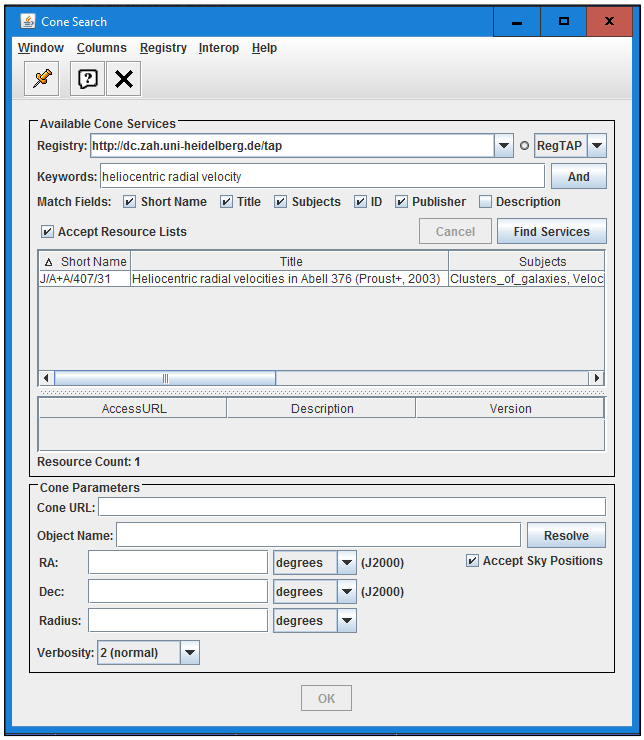


Figure 7: Résultat d'une recherche de services par TopCat, (v 4.4)

Toutefois, cette recherche doit être répétée pour chaque annuaire, et pour chaque type de données recherché. De plus, les mots-clés renseignés dans les services ne sont pas forcément les mots-clés utilisés instinctivement par l'utilisateur ; ce qui nuit à la fiabilité de la recherche de services. De plus, et quelle que soit la méthode utilisée les services susceptibles de retourner des données correspondant à ces critères de sélection sont potentiellement nombreux (plusieurs centaines), et le choix final des services sélectionnés reste à la charge de l'utilisateur. La Figure 8 illustre cette difficulté, avec 236 services disponibles à tester simplement en relaxant une contrainte par rapport à la requête précédente (le mot-clé demandé est ici « radial velocity » au lieu de « heliocentric radial velocity »).

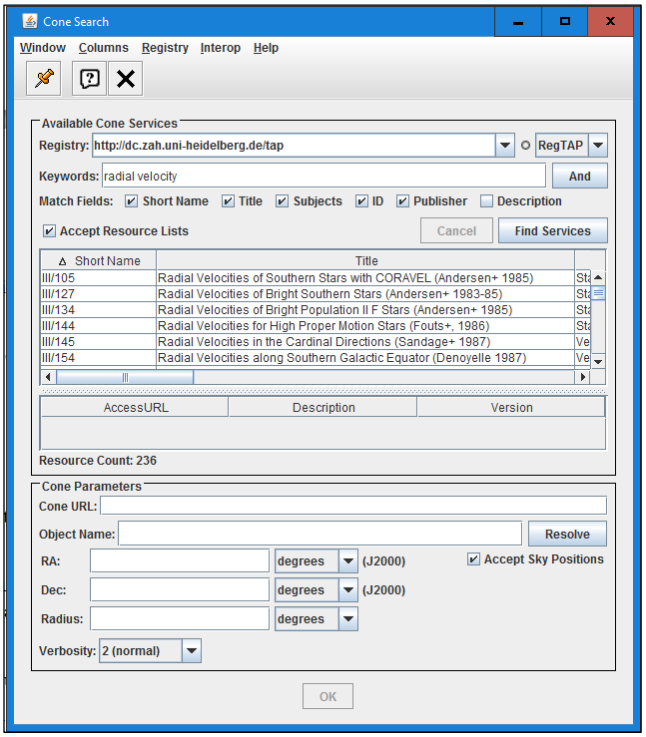


Figure 8: Résultat d'une requête dans TopCat (v 4.4) en relaxant une

B) ALADIN

Aladin est probablement le logiciel « compatible OV » le plus utilisé, en raison des multiples possibilités de traitement, d'affichage et de consultation de services qu'il propose. La sélection de services dans Aladin effectue un premier niveau de filtrage en ne proposant de retourner que les services capables de répondre à une requête formulée pour un objet particulier. La Figure 9 illustre cette possibilité, l'interface de gauche affichant les services interrogés qui répondent positivement à une requête concernant l'objet « Aldebaran ». Aladin propose également un filtre des services par mots-clés, mais ces mots-clés ne peuvent provenir que des noms de services. La Figure 9 réutilise le mot-clé « radial velocity » interrogé dans Topcat, pour lequel aucun service n'est proposé dans Aladin sur l'interface de droite.

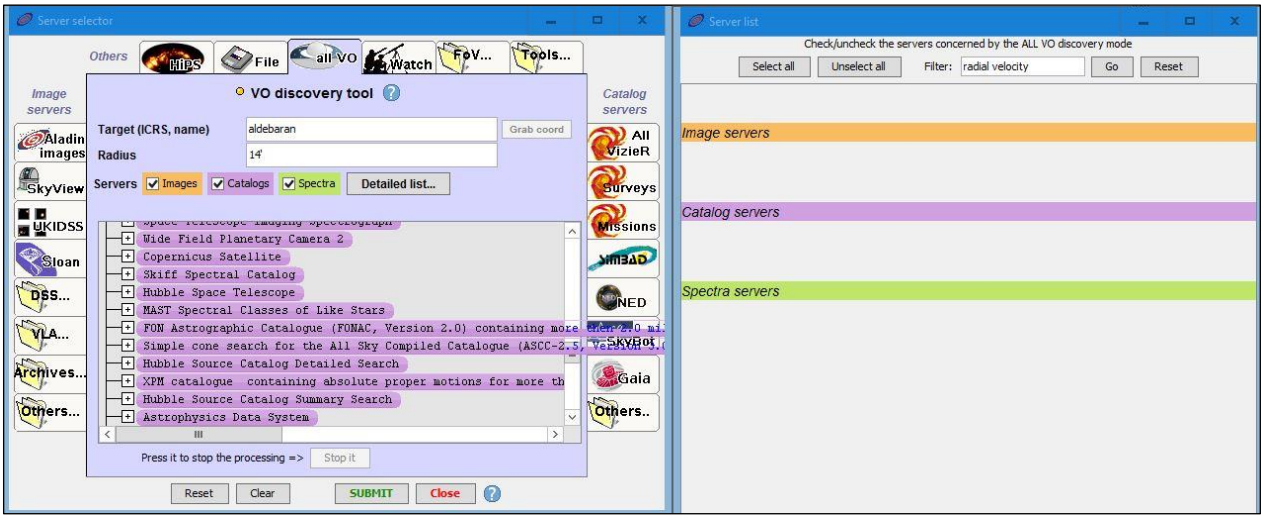


Figure 9: Mots-clés et recherche de services dans Aladin (v9.013)

L'interface de choix de services d'Aladin, présentée en Figure 10 impose à l'utilisateur d'opérer un tri manuel, à partir des descriptions générales des services accessibles en utilisant le bouton « ? ». Les descriptions des informations disponibles dans les services ne sont pas accessibles directement, pour les obtenir il est nécessaire d'envoyer une requête à chacun des services. Le choix final du ou des services à utiliser pour obtenir une information précise est donc difficile. De plus, le nombre déjà important de services proposé par Aladin (669 services pour les catalogues d'observations, sans compter les spectres ni les images) ne reflète qu'une partie des services OV disponibles, qui sont plus de 10000 pour les catalogues.

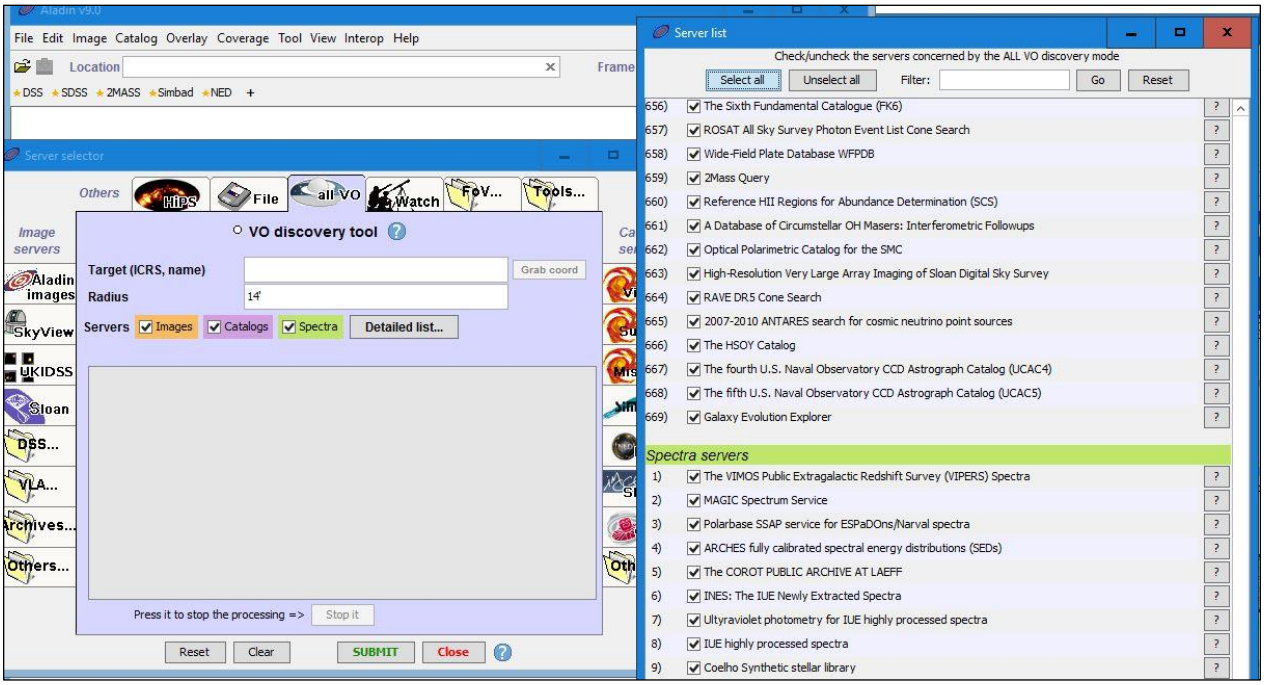


Figure 10: Interface de choix de services dans Aladin (v9.013)

C) LOGICIELS ET SELECTION DE SERVICES

Comme nous venons de le voir sur deux logiciels très utilisés dans l'OV, le choix des services à interroger pour obtenir une information est faiblement guidé. En particulier, parmi les services indiqués par ces outils le choix final n'est pas automatique et s'opère donc :

- soit par tâtonnements en essayant différents service et en sélectionnant ceux qui retournent effectivement des données correspondant aux besoins de l'utilisateur
- soit en utilisant une connaissance apriori des services disponibles (que l'utilisateur ne possède pas toujours, et rarement pour l'intégralité des services disponibles)
- soit par habitude en utilisant un ensemble de services aux profils bien connus.

1.3.5. Ontologies et astrophysique

La description des objets étudiés en astrophysique est un problème complexe. En fonction des multiples catalogues, des catégories de classification utilisées et des profils instrumentaux utilisés pour les observations, les caractéristiques de ces objets peuvent se trouver disséminées dans différentes sources d'informations. Des services automatiques de classification et de restitution d'information existent (SIMBAD[42] du CDS de Strasbourg est parmi les plus connus et les plus utilisés), mais des problèmes d'hétérogénéité des informations (objets identifiés suivant les services comme galaxies ou comme étoiles, descriptions hétérogènes et parfois incomplètes...) demeurent.

Même une classification simple des sources par leur nom pose le problème de la multiplication des catalogues et donc des références différentes pour un même objet, d'où l'utilité d'une classification par types d'objets complémentant une classification par identifiants. Une approche à base d'ontologies a été envisagée (Derriere et al. 2010) pour régler ces types de problèmes et rapprocher les descriptions internes aux annuaires OV interrogés de types d'objets connus par ailleurs. La base de données utilisée pour la construction de cette ontologie a été SIMBAD.

Une initiative (Thomas 2015) a été conduite dans le but d'utiliser une ontologie pour permettre de retrouver des ressources dans un des annuaires de l'IVOA en utilisant les mots-clés les plus pertinents par rapport à la description de l'utilisateur. Cette ontologie se propose de décrire des « sujets » (système solaire, étoiles brillantes…) en rapport avec le contenu des services inscrits dans les annuaires.

[42] http://simbad.u-strasbg.fr/simbad/

Pour Helio, organisant un OV dans le domaine de l'héliophysique les préoccupations sont voisines, mais l'intégration d'une ontologie dans le système d'Observatoire Virtuel est plus avancée. L'ontologie d'Helio(Bentley et al. 2013) est au cœur d'une des briques de cette architecture, le SMS (Semantic Mapping Service) permettant aux différents modèles de données, dictionnaires et types de fichiers de se rejoindre sur les concepts qu'ils ont en commun et qui sont décrits dans l'ontologies utilisée. Helio a modélisé aussi bien les concepts issus de la connaissance physique du domaine que les systèmes de coordonnées utilisés et jusqu'aux structures humaines (observatoires, personnes et leur rôle dans les équipes...).

Le niveau d'abstraction qu'il est possible d'exprimer dans les ontologies en ont fait un champ d'investigation en astrophysique, au-delà des OVs. Pour l'instrument MAIA(Pessemier et al. 2015) destiné au télescope Mercator, deux niveaux d'abstractions (un niveau général, méta-modèle de l'instrument et un niveau plus technique décrivant les applications concrètes) couvrant de nombreux aspects technologiques de l'instrument ont été modélisés de cette façon . Les auteurs de ce travail concluent que l'utilisation d'ontologies pour modéliser leur instrument a permis une communication plus efficace entre les agents logiciels de contrôle et une maintenabilité plus grande. Comme il est mentionné dans ces travaux, la représentation sémantique des capacités d'un instrument peut conduire à la collaboration entre différents instruments. Et une telle conception peut également bénéficier à la représentation ontologique des données. À condition que les problèmes d'alignement puissent être résolus, cela pourrait permettre l'échange de données et la mise en correspondance des services astrophysiques avec le système de contrôle des instruments. Cela permettrait par exemple d'identifier, dès la réalisation des observations, des données complémentaires aux données observées ; et la mise à disposition immédiate des données observées dans un schéma partagé par plusieurs acteurs serait également envisageable.

1.3.6. Composition de services en astrophysique

L'Observatoire virtuel est la source la plus importante de données astrophysiques disponibles. L'interopérabilité des données au sein des services OV est partiellement assurée par l'utilisation d'un vocabulaire dédié (UCDS, UTYPES) et l'utilisation de formats communs. Les services VO peuvent être trouvés en interrogeant les registres OV jouant le même rôle que les registres UDDI; Grâce à l'utilisation de mots-clés qui mèneront à des descriptions XML des services qui peuvent enfin être exécutés par un logiciel compatible OV.

L'astrophysique a une certaine spécificité, qui lui vient de son statut de science théorique et observationnelle, comme du fait que ses instruments constituent une source constante de données massives et nativement hétérogènes. La double nature de science théorique et de science d'observation conduit à une description hétérogène de la même sémantique. Un spectre théorique, décrivant les mêmes quantités qu'un spectre observé aura une description en partie différente. En particulier, un spectre calculé pour un profil générique d'étoile ne contiendra par définition pas de nom d'objet observé, ou de coordonnées sur le ciel. Un spectre théorique peut ne pas être décrit par les mêmes

paramètres qu'un spectre résultant d'une observation, même si leur sémantique (c'est à dire le sens qu'ils portent et les informations qu'ils contiennent) est la même.
L'apparition de nouveaux instruments conduit à l'apparition de nouveaux services, entraînant de nouvelles façons de décrire les résultats, ou à de nouvelles gradations dans le niveau de détails disponibles. Potentiellement, cela augmente encore l'hétérogénéité des données disponibles dans les services. Les termes généraux propres à la discipline peuvent être déclinés en sous-ensembles plus précis, eux-mêmes décrits parfois de manières différentes selon les usages de leur communauté scientifique d'appartenance ou les besoins de leurs instruments originaux. Il en va de même pour les unités et les formats de données.

De plus, chaque instrument possédant sa propre niche scientifique, les cibles observées diffèrent beaucoup d'un instrument à l'autre, même pour les instruments ayant des caractéristiques similaires. Par conséquent le fait qu'un service est susceptible de fournir des informations bien définies, ne signifie pas que le service en question contienne effectivement une information utile pour une cible spécifique. Les services sont donc hétérogènes en description, et le résultat de leurs requêtes sera hétérogène en contenu. Bien que l'OV fournisse une couche d'interopérabilité pour unifier ces données hétérogènes, le contenu réel de la description des services et du contenu dépend donc en grande partie du contexte. En conséquence, il est parfois nécessaire d'adapter les formats OV eux-mêmes selon les sous-champs à décrire, comme l'astrochimie dans VAMDC (Dubernet et al. 2016).

Ces difficultés expliquent que la composition des services pour l'astrophysique est assurée principalement par des logiciels très spécifiques et dépendants du contexte, intégrés dans les «Science Gateways» (SGN) qui font aujourd'hui partie intégrante de la conception d'un instrument. L'instrument à venir Cherenkov Telescope Array (CTA) offre un tel exemple (Costa et al. 2015). La seule approche pour la composition des services en astrophysique disponible en-dehors des SGN est AstroTaverna (Ruiz et al. 2014), basé sur le gestionnaire de workflow Taverna. Cette plate-forme s'adresse à des utilisateurs possédant une formation technique suffisante pour définir eux-mêmes des compositions de services. Même si les compositions résultantes sont partagées par l'utilisation d'un serveur Web dédié, la modification des workflows existants reste une tâche difficile. AstroTaverna s'appuie sur les normes OV disponibles pour décrire et découvrir les services mais n'inclut pas de couche sémantique qui aiderait à uniformiser la description des services. En outre, la sélection finale du service est à la charge de l'utilisateur et ne propose pas de classement a priori, ni sur la qualité des services ni sur la disponibilité d'une information précise dans les services (par exemple des observations d'un objet donné).

1.4. Positionnement et problématique

1.4.1. Positionnement

Le contexte applicatif de ce travail de thèse est celui de l'astrophysique, et le problème posé dans ce contexte est celui de la récupération ou de la génération automatique d'informations par le biais de la composition de services. Plusieurs problèmes rencontrés par les astrophysiciens pour mettre en relation

des données de provenance différente, et en utilisant le maximum de fournisseurs de traitements et de données disponibles restent à résoudre.
Nous avons vu que l'OV propose des solutions de découverte de services, et vise à une relative homogénéité de ces services. Toutefois, des opérations coûteuses en temps et exigeantes en compétences restent à la charge de l'utilisateur final (sélection des services, vérification des contenus précis). Les mots-clés utilisés pour la recherche de services ne correspondent pas automatiquement aux habitudes de tous les utilisateurs. Lorsqu'ils correspondent, nous avons vu que les interfaces de recherche de services actuelles peuvent sélectionner un grand nombre de services. Devant la difficulté à trier parmi plusieurs centaines de services quels sont ceux susceptibles de fournir des informations utiles pour son cas scientifique, un astrophysicien se dirigera d'abord vers les services qu'il connaît déjà. Cette difficulté est augmentée lorsque plusieurs informations susceptibles de provenir de services différents sont recherchées.
De plus, l'OV décrit uniquement des services de données, et pas de service de traitement ou d'analyse de ces données. Il n'existe aucun mécanisme de médiation et d'orchestration de services proposé par l'OV. Lorsque des informations pourraient être obtenues par un enchaînement de services différents, rien ne permet de détecter cet enchaînement.
Enfin, l'OV n'est pas monolithique, et plusieurs variations existent (VAMDC, IPDA…). Rechercher des informations dans des services issus d'implémentations différentes exige de s'adresser à des modèles de données différents, bien qu'issus de la même architecture et demanderait en conséquence à interroger un méta-modèle qui n'est pas spécifié. L'utilisation concrète, quotidienne et systématique de l'Observatoire Virtuel reste donc compliquée y compris pour un utilisateur averti en raison de ces difficultés.

Néanmoins, l'OV fournit une infrastructure permettant déjà un certain niveau d'interopérabilité. Ce niveau d'interopérabilité peut être évalué à travers un framework issu de travaux pour l'interopérabilité des entreprises conduits dans le cadre du réseau d'excellence européen INTEROP (Nieto 2011) dont une représentation est illustrée en Figure 11.

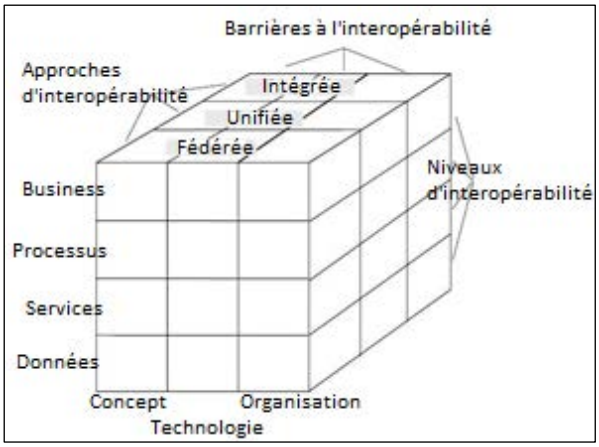


Figure 11: Framework d'interopérabilité des entreprises (Nieto 2011)

Ce framework sépare trois aspects de l'interopérabilité :

- Sa portée , qui peut se situer au niveau des données, des services (services de l'OV pour l'astrophysique) et des processus(enchaînements de services).
- L'approche utilisée. Intégrée lorsqu'un format commun existe pour tous les modèles, unifiée lorsque ce format commun n'est présent qu'au niveau de méta-modèle non implémentable mais permettant la reconnaissance entre entités issues de ce méta-modèle. L'approche fédérée s'organise sans partage de modèle, de langage ni de méthode et implique l'intervention d'une ontologie pour réunifier les concepts.
- Les barrières qui peuvent être conceptuelles (différences de syntaxe et de sémantique entre données échangées), technologiques (protocoles d'échange, standards de données) ou organisationnelles (droits et responsabilités de chacun, organisation hiérarchique des parties prenantes).

La couche « Business » est trop éloignée des préoccupations du contexte, et les aspects organisationnels n'ont que peu d'influence dans le contexte astrophysique. Néanmoins les couches données, services et processuscomme les deux autres barrières et les trois approches restent pertinentes.

L'interopérabilité actuelle dans le contexte de l'astrophysique est représentée à gauche de la Figure 12, alors que l'approche vers laquelle nous souhaitons travailler se situe à droite de la même figure.

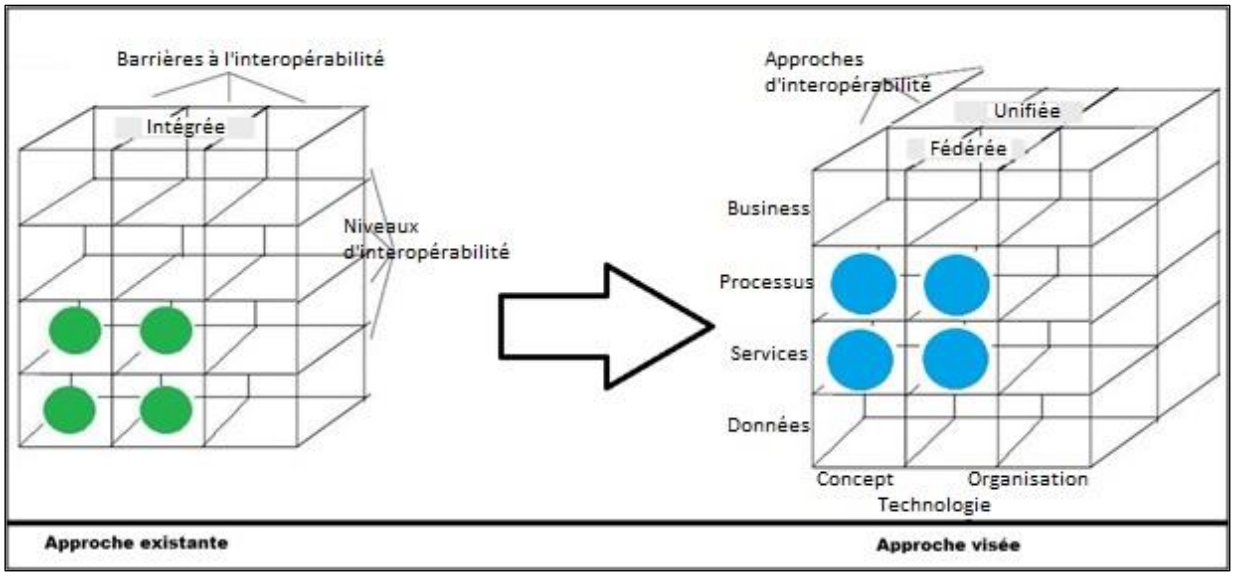


Figure 12: Niveaux d'interopérabilité en astrophysique

Nous souhaitons donc passer d'une interopérabilité intégrée au niveau des services et des données à une interopérabilité fédérée au niveau des services et des process. Cela permettra de faire collaborer entre eux des services issus de l'OV et des services qui n'en sont pas issus, indépendamment de leur profil (services fournisseurs de données ou de traitements, bases de données ou bibliothèques logicielles…). Cela permettra également d'envisager des enchaînements de services (process) eux-mêmes indépendants des plateformes d'implémentation des services.

1.4.2. Problématique

La problématique adressée dans ce manuscrit consiste à trouver des solutions pour contribuer à atteindre ce niveau d'interopérabilité. Nous proposons d'étudier les possibilités d'extension de l'interopérabilité actuelle grâce à l'utilisation de la représentation sémantique des connaissances contenues dans les services disponibles, pour lever les barrières conceptuelles et technologiques existantes. Une approche fédérée passe par une représentation ontologique du contenu des services. Puisque les ontologies constituent l'élément de base de cette architecture, il est nécessaire de préciser les contours des recherches menées dans ce domaine. Les principaux axes de recherche au sujet des ontologies dans les sciences de l'information sont :

- L'élaboration de standards (Smith et al. 2007; Hucka et al. 2015)
- L'élaboration de méthodes de conception d'ontologies (Baccigalupo & Plaza 2007; Haase et al. 2008)
- L'alignement d'ontologies (Shvaiko & Euzenat 2013)
- La population d'ontologies (Pia 2015). Les recherches actuelles en informatique visent à produire des algorithmes de population d'ontologies automatiques, ou semi-automatiques nécessitant la validation et la supervision d'un opérateur humain.
- La composition de services Web sémantiques (Bansal et al. 2014; Puttonen et al. 2013; Rodriguez-Mier et al. 2016)

Des difficultés de conception automatique d'ontologies, de peuplement et de composition automatique de flux de traitement se posent, qui ne sont pas définitivement résolus par les recherches actuelles. Ces difficultés sont rencontrées notamment lorsque :

- Peu de représentations structurées de la connaissance du domaine peuvent être exploitées pour la définition de l'ontologie
- L'extraction d'informations et l'analyse du langage naturel ne peuvent s'appuyer que sur des textes courts, non structurés, très spécialisés et de niveaux de détails variables

De plus, Les ontologies de services existantes sont peu adaptées pour la description de services exprimés en-dehors des formats connus dans l'industrie (SOAP, WSDL, UDDI). De plus, elles prévoient uniquement la description de services Web. Les services dont il est question dans ce mémoire ne sont pas uniquement des services Web, mais peuvent également être des bibliothèques de traitement, ou des logiciels scriptables par exemple. Une structure de description adaptée devra donc être proposée.

1.5. Conclusion

Les spécificités de l'IVOA, qui s'écarte des standards des sciences de l'information, rendent en grande partie inopérantes les recherches en informatique qui pourraient contribuer à une amélioration de l'efficacité globale de son architecture. La composition automatique de services et la transition vers le Web sémantique sont notamment rendues plus difficiles. De plus, les ressources disponibles concernant le domaine de l'astrophysique dans les ontologies utilisées habituellement pour la

conception automatique et la population d'ontologies sont faibles, et peu d'ontologies dédiées au domaine sont disponibles. Les descriptions précises des capacités des services (les quantités contenues dans les données qu'ils hébergent) sont exprimées en langage naturel pour les astrophysiciens, c'est-à-dire par l'utilisation de vocabulaire très spécialisé et sans forme grammaticale.

Ces difficultés peuvent être traduites en exigences, qu'il faudra satisfaire afin de proposer des solutions permettant la représentation et la composition automatique de services. Ces exigences sont de deux ordres :

- Les exigences fonctionnelles, dont la satisfaction permet l'utilisation des services. Un exemple d'exigence fonctionnelle consiste à pouvoir identifier les combinaisons d'entrées nécessaires pour pouvoir invoquer un service.
- Les exigences non fonctionnelles, qui sont à satisfaire afin d'assurer la meilleure utilisation possible des services. Un exemple d'exigence non fonctionnelle est la capacité d'identifier le meilleur service possible pour l'accomplissement d'une tâche ou l'obtention d'une information donnée.

Dans la suite de ce manuscrit, nous verrons quels sont les problèmes théoriques posés par ces exigences, et nous proposerons des solutions à ces problèmes théoriques. Au centre de ces solutions théoriques, nous trouverons la définition, l'enrichissement et la population d'une ontologie de services. Nous examinerons les conséquences du profil de cette ontologie et des services qu'elle contient sur la composition automatique de services. Nous proposerons des algorithmes pour parvenir à une composition de services automatique, performante et facilement paramétrable. Le chapitre suivant expose la structure de l'ontologie de services que nous proposons, et les systèmes mis en œuvre pour représenter les spécificités des services que nous avons à décrire.

Chapitre 2: Un module ontologique générique pour les services

2.1. Introduction

La description des services Web par l'utilisation d'ontologies n'est pas un sujet récent, et des études portant sur l'avancée de la recherche dans ce domaine sont disponibles (Tosi & Morasca 2015). Ces études soulignent que les cas d'utilisation réels ne sont pas faciles à trouver, même dans la littérature récente. La plupart des travaux dans le domaine restent théoriques, mais des orientations pour des recherches futures sont proposées: *«Mettre l'accent sur la définition et l'adoption d'un formalisme et d'un langage standard de facto pour créer des ontologies pour différents domaines. Plusieurs méthodologies et langages existent déjà, mais leur compatibilité et leur interopérabilité ne sont pas prouvées et loin de la réalité. »* (Tosi & Morasca 2015).

Dans ce chapitre, nous aborderons certaines de ces orientations en proposant un formalisme décrivant des services correspondant à un profil particulier. Nous utiliserons et illustrerons ce formalisme à partir des services astrophysiques. L'amélioration de la sélection et de l'interopérabilité des services, passe par une description commune de leur domaine d'application. Cette description doit être complétée par des concepts simples permettant d'utiliser concrètement les services décrits en assurant la compatibilité des résultats qu'ils produisent.

Nous verrons en conséquence comment constituer l'ontologie proposée dans ce manuscrit en fonction des particularités des services que nous avons à traiter. Nous présenterons la méthodologie utilisée pour cette construction, et détaillerons les particularités des services et leurs implications sur la représentation de ces services. Puis nous exposerons les propositions que nous faisons pour en tenir compte convenablement, afin de proposer une ontologie de services résolvant les problèmes théoriques que ces particularités soulèvent.

2.2. Etat de l'art

Afin de bien cerner les spécificités des services rencontrés dans notre cas d'application, nous allons détailler certains aspects des technologies de services Web existantes. Nous évoquerons ensuite les approches d'enrichissement sémantique de ces technologies et situerons le cas des services traités dans ce manuscrit par rapport à ce panorama.

Une des technologies de services Web les plus utilisées est l'architecture REST. Cette architecture est proposée dans la thèse de Thomas Fielding (Fielding 2000). Cette architecture d'appel de services à distance tire les leçons de l'étude de plusieurs architectures d'applications basées sur le réseau. Elle reprend certains principes issus des approches précédentes (Code-on-demand, communication client-serveur, dialogue à étape unique sans états) et propose une interface d'échanges unifiée, indépendante du domaine d'application. Le domaine spécifique est décrit dans les métadonnées du message échangé entre le client et le serveur, ainsi que dans les données de contrôle (Control data) présentant notamment l'action à effectuer et les paramètres de requête. Les concepts énoncés dans la définition de

l'architecture REST sont indépendants de leur application technique. Toutefois, dans son document Thomas Fielding illustre ces concepts en se basant sur l'architecture Web et propose un cas d'application utilisant les composants du Web (Protocole http, serveur Apache, utilisation d'URI pour la localisation des ressources…). Aujourd'hui, les termes d'application Web (ou API Web), API REST et RESTful services sont des synonymes utilisés presque exclusivement pour désigner des applications client/serveur utilisant les ressources du Web pour échanger leurs messages et décrire les données et la signification des messages échangés.

Web Services Modeling Ontology (WSMO) et Web Language Ontology for Services (OWL-S) sont des ontologies spécifiquement conçues pour décrire des services Web. Indépendantes des technologies utilisées pour accéder aux services, elles utilisent un système d'appels des services (le « *Grounding* » de OWL-S, et l'élément « *Choreography* » de WSMO) qui renvoie notamment vers les spécifications de format des messages d'entrée et de retour attendus et renvoyés par les services. Qu'il s'agisse de OWL-S ou de WSMO, la description technique du service (le format de requête, les paramètres etc.) se base en grande partie sur WSDL.
Toutefois, la définition d'ontologies spécifiquement dédiées à la représentation de services Web n'est pas la seule alternative étudiée pour amener de la représentation sémantique à ces services. D'une part, via l'élément « wsdl :description », WSDL permet nativement l'inclusion d'éléments sémantiques définis par exemple en utilisant le langage OWL. D'autre part, une extension de WSDL appelée WSDL-S (Akkiraju et al. 2005) permet d'inclure l'utilisation d'éléments sémantiques extérieurs, directement dans les définitions WSDL des services Web. Ces éléments sémantiques peuvent par exemple être définis dans des fichiers (.xsd) qui sont alors référencés dans les définitions WSDL des services et peuvent servir à préciser les éléments sémantiques des services. WSDL-S, en externalisant la définition des aspects sémantiques permet non seulement d'alléger les définitions WSDL qui comprendraient un grand nombre de ces éléments, mais également de faire référence à des éléments en assurant que leur définition soit commune à toutes les descriptions. La logique de cette approche a été poursuivie dans une recommandation du W3C appelée SAWSDL (Farrell & Lausen 2007), qui propose de définir des balises dans les documents WSDL pour référencer directement des concepts présents dans des ontologies externes.
Définir ces ontologies externes de telle façon qu'elles expriment correctement la sémantique des services est au cœur de la définition de WSMO-Lite (Roman et al. 2015), qui est une ontologie de services construite à partir de WSMO. WSMO-Lite utilise le langage OWL et ne reprend pas le langage WSML, spécifique à WSMO. Elle évacue également tous les aspects de WSMO qui ne sont pas directement liés aux services, comme les ontologies externes et les mediators. Bien que WSMO soit très liée à WSDL, les auteurs de WSMO-Lite présentent la volonté d'unifier services RESTful et services conformes à WSDL autour de concepts communs dans une ontologie de modélisation de service unique. Ils exposent en conséquence dans le même article le mécanisme d'annotation sémantique des APIs RESTful hRESTS qui permet d'ajouter des annotations sémantiques directement dans les pages html pour les services RESTful. Ces pages annotées sont ensuite traitées comme des descriptions WSDL et les services peuvent alors être décrits par WSMO-Lite. Ce mécanisme exige une description préalable du service Web considéré, comprenant tous ses aspects fonctionnels (méthode

d'invocation GET/POST, URI, nom des paramètres, des opérations possibles, des entrées et des sorties…)

OWL-S et WSMO sont des ontologies de haut niveau d'abstraction. Les connaissances du domaine d'application des services qu'ils décrivent est hors du champ d'application de leurs spécifications. Cela signifie que les descriptions ayant trait au domaine d'application des services Web doivent être convertis dans une expression ontologique commune, ou au moins dans des expressions interopérables. Un tel travail est non trivial et doit être fait au cas par cas (Bensaber & Malki 2012). Ces ontologies s'appuient sur deux hypothèses : elles décrivent des services Web, et ces services Web s'expriment dans un format connu (WSDL ou une API REST au profil documenté) sur laquelle s'appuie sa description sémantique. Elles ne couvrent pas le cas de profil de services Web ne s'inscrivant ni dans une implémentation du standard REST bien définie ni dans une description WSDL. Elles ne sont pas non plus conçues pour décrire des services différents des services Web, tels que des bibliothèques logicielles, des bases de données ou des invocations de scripts par exemple. En raison du fait que WSMO s'appuie sur différentes couches d'ontologies pour décrire les services et couvre certains aspects (par exemple les médiateurs) que OWL-S ne couvre pas, il est plus compliquée de décrire un service dans WSMO que dans OWL-S. Un exemple théorique d'un service décrit dans WSMO avec les explications correspondantes peut être trouvé dans la documentation de WSMO[43]. En dehors de cette documentation WSMO manque de mises en œuvre concrètes, ce qui est également le cas de OWL-S (Kamaruddin et al. 2012).

Dans notre cas d'application astrophysique, l'architecture de l'IVOA définit les protocoles d'accès aux données. Ces protocoles sont spécifiques à cette architecture, et les services qu'ils servent ne sont pas exprimés par le langage WSDL. En conséquence, la description de l'appel de ces services par les concepts proposés par OWL-S ou la chorégraphie de WSMO est difficile. L'architecture REST est la plus approchante des profils de l'architecture de l'IVOA. Les définitions des métadonnées échangées est toutefois variable suivant les services, et dépend du modèle de données utilisé par le service, défini par un schéma XML spécifique. Les données de contrôle dépendent du protocole utilisé, bien que ces protocoles soient soutenus par le protocole HTTP. La description d'un service de l'IVOA ne fait pas mention de ses paramètres d'entrée, mais référence un protocole d'accès par lequel ces paramètres sont fixés. La sémantique de ces paramètres d'entrée ne peut donc pas être déduite de la description du service, de plus l'URL à utiliser pour contacter le service ne peut pas être formée. En effet, les URIs contenues dans la description de services ne contiennent ni la représentation technique des paramètres de requête ni leur sémantique. La représentation technique des paramètres, comme leur sémantique sont contenues dans la spécification des protocoles utilisés par les services. Précisons que, sémantiquement les entrées et les sorties des services de l'IVOA ne sont pas des concepts ou des individus isolés. Ils résultent de la combinaison de plusieurs individus, étant donné qu'une quantité astrophysique exprime une sémantique particulière associée à une unité et un format. En outre, il peut

[43] http://www.w3.org/Submission/WSMO-primer/

être nécessaire que certaines de ces combinaisons proviennent d'une source unique lorsqu'elles sont traitées comme entrées pour un service.

Les bibliothèques scientifiques sont des logiciels locaux, et leur utilisation concrète (protocole d'utilisation, mise en forme des requêtes…) n'a rien de commun avec celle d'un service Web. Nous devons donc fournir un mécanisme simple d'utilisation, (« *grounding* ») permettant de décrire correctement les protocoles venant d'une éventuelle *Distributed Computer Infrastructure* (DCI) préexistante, les services Web indépendants et les bibliothèques locales. Le « *Grounding* » des services est par conséquent une préoccupation primordiale, car nous voulons pouvoir utiliser réellement toutes les sources d'information et de traitement de l'information disponibles dans l'ontologie de services qui sera au centre de notre système.

Néanmoins, même avec un *grounding* adapté la mise en forme des entrées dans le message envoyé au service et l'interprétation du contenu du message de résultat peut être délicat ; en raison de la typologie de la connaissance du domaine elle-même. Tout d'abord, chaque entrée correspond dans notre cas d'utilisation à une quantité qui peut être exprimée de multiples façons. Deux spectres semblables, par exemple, peuvent être exprimés en diverses combinaisons d'unités et de formats par différents services. Ils peuvent également être acceptés pour des combinaisons différentes d'unités et de formats dans les entrées de différents services. Ainsi, les relations «*hasInput*» et «*hasOutput*» définies dans OWL-S doivent être liées à un mécanisme généralisant ces exigences, en indiquant quelles unités et formats sont associés à quelle mesure pour chaque entrée et chaque sortie de service. En outre, certains de ces paramètres peuvent être corrélés, en ce sens qu'ils doivent provenir des sorties d'un seul service. Cela se produit par exemple lorsqu'il est nécessaire de disposer d'une mesure et des barres d'erreur sur cette mesure. Les barres d'erreur et la mesure doivent exister en tant que paramètres dans l'ontologie et leur disponibilité doit être résolue conjointement lors de la phase de sélection de services. Plus généralement, cela se produit dès que deux ou plusieurs mesures données peuvent avoir une influence les unes sur les autres, sont interdépendantes ou doivent être prises dans les exactes mêmes conditions d'observation.

Par conséquent et malgré leur utilité, ni OWL-S ni WSMO ou WSMO-Lite ne couvrent entièrement les exigences de tous les champs d'application d'une manière concrète et utilisable. C'est ce qu'illustre ce manuscrit en prenant le cas d'application astrophysique. Notre but est de construire une ontologie de services plus souple que OWL-S concernant les mécanismes d'appel de services et la description des paramètres d'entrée / sortie. Nous visons également à éviter l'approche multicouche de WSMO et l'utilisation d'un langage spécifique, comme WSML pour WSMO.

2.3. Le schéma d'ontologie proposé

L'état de l'art a soulevé certaines limites des ontologies de services actuelles pour notre cas d'application. Pour repousser ces limites, nous proposons une ontologie de services dont la structure sera adaptée au profil des services rencontrés dans notre domaine d'application. Nous intitulons le schéma d'ontologie proposé dans ce manuscrit « ASON » pour « *Astrophysical Services ONtology* »,

puisque nous utilisons le cas d'application astrophysique pour illustrer l'utilisation de ce schéma et préciser les impératifs qui ont guidé sa conception. Toutefois, son utilisation n'est pas captive d'un domaine particulier et pourra être envisagée dans tous les cas d'applications présentant les mêmes spécificités que celles décrites dans ce document et illustrées par l'astrophysique.

Pour pouvoir assurer une interopérabilité sémantique entre des services, la description sémantique des services proprement dits (c'est-à-dire leurs aspects fonctionnels tels que les entrées et les sorties, les protocoles et adresses, …) doit s'accompagner d'une représentation sémantique de leur domaine d'application. Cette représentation du domaine d'application permet de réunifier les descriptions des concepts non fonctionnels entre les services. De cette façon la sortie d'un service est par exemple exprimée à l'aide d'un concept dans une ontologie de domaine, et rapprochée d'un concept voisin dans la même ontologie ou dans une ontologie différente. Ce concept voisin peut être l'entrée d'un autre service et indiquer un enchaînement possible, ou bien également une sortie par exemple, et indiquer que les capacités des deux services se recouvrent en partie. Lors de l'utilisation des ontologies de services existantes, cette réunification des concepts non fonctionnels n'est pas proposée nativement. Pour être en mesure d'obtenir une description partagée du domaine d'application couplée à une description fonctionnelle des services, nous proposons une ontologie modulaire. La structure modulaire de cette ontologie vise à séparer la description des aspects techniques des services, de la représentation de leur domaine d'application.

ASON se compose donc de deux modules, le premier qui s'intitule « *GEneric Ontology for Services* » (GEOS) et le second est un module thématique qui, pour le cas astrophysique s'intitule « ASTRO-THEM ». ASON est une ontologie globale (Wache et al. 2001), qui contient tous les services qu'elle décrit accompagnés d'une représentation de la connaissance liée à leur domaine d'application. Cette caractéristique permet à ASON d'assumer le rôle d'un annuaire de services, en même temps que celui d'une ontologie de services et d'une ontologie de domaine. En comparaison, les approches existantes imposent l'utilisation d'un annuaire de services en plus d'une réconciliation sémantique entre les capacités des services et la description du domaine d'application (Bansal et al. 2014).

Le module GEOS a pour rôle de décrire les aspects techniques des services, indépendamment de leur domaine d'application. La définition de ce module répond à la nécessité de prendre en compte certaines spécificités de services qui rendent difficile l'utilisation des ontologies de services existantes. Sa tâche est d'exprimer les exigences fonctionnelles des services, afin que ces exigences puissent être satisfaites lors de la composition de services présentée au chapitre 4. Ce chapitre examinera ces exigences fonctionnelles, les difficultés de représentation qu'elles posent et les solutions apportées par GEOS.

.

2.4. Méthodologie de développement d'une ontologie

Plusieurs méthodes pour la création et la maintenance des ontologies existent, et ont été discutées dans des contextes spécifiques d'application comme l'industrie (Karray et al. 2012) ou d'un point de vue plus général (López 1999). Un panorama dressé plus de dix ans après le précédent (Dwyer 2013) pointe

le fait que peu de nouvelles méthodologies émergent, comme les difficultés rencontrées pour faire la preuve de l'efficacité de chacune des approches existantes. La plupart d'entre elles sont des résumés organisés de bonnes pratiques issues d'expériences apprises par la pratique du développement d'ontologies.

En particulier, un des critères d'analyse commun aux deux panoramas généraux cités ci-dessus concerne l'identification des concepts qui formeront la taxonomie de l'ontologie. Pour ce critère, (Dwyer 2013) regrette le manque d'originalité des approches considérées dont la plupart reposent sur des méthodes anciennes du type de celle proposée en 1996 (Uschold et al. 1996), dans la partie « Ontology Capture ». Des approches pour la construction et la population automatique ou semi-automatique d'ontologies ont été proposées comme OntoCASE (Blomqvist 2009) depuis, de même que des méthodes d'évaluation comme OntoQA (Tartir et al. 2005) ou LOVMI (Richard et al. 2015) , sans que pour l'heure les méthodologies organisées couvrant tout le cycle de vie des ontologies ne les reprennent à leur compte.

Nous choisissons d'utiliser l'approche METHONTOLOGY (Baccigalupo & Plaza 2007), qui est une des méthodologies les plus anciennes mais aussi une des mieux documentées disponibles. Les activités techniques décrites dans METHONTOLOGY vont de la spécification à la maintenance en passant par la conceptualisation, la formalisation et la mise en œuvre. Les activités de soutien sous-tendent les activités techniques, afin de s'assurer qu'aucun aspect du développement ne soit négligé. Une vue générale de METHONTOLOGY est disponible dans la figure 13.

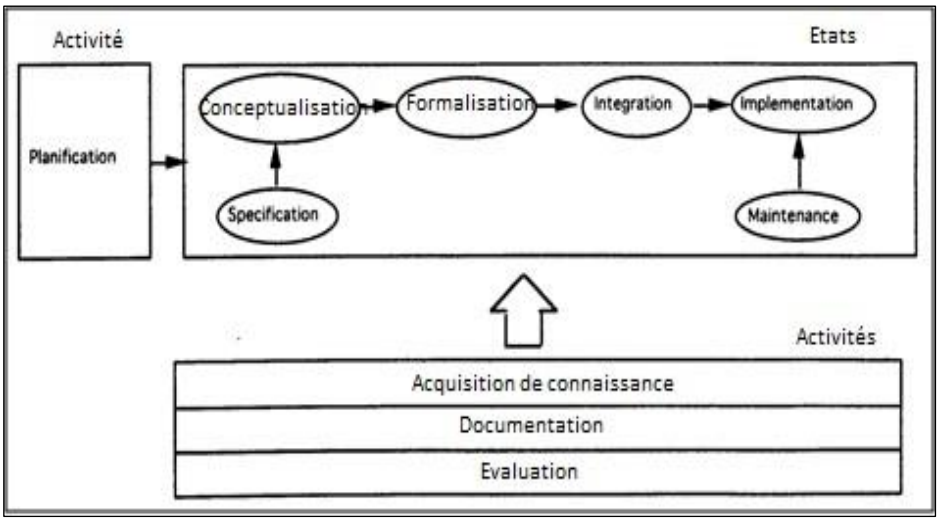


Figure 13: Vue générale de la méthode METHONTOLOGY (Fernández-López et al. 1997)

Les « *States* » sont les étapes par lesquelles passent le cycle de vie de l'ontologie, de sa conceptualisation à son implémentation. Les « *activities* » identifient les actions à mener pour assurer le bon déroulement de chacune de ces étapes. Les sections suivantes décriront le développement de l'ontologie au centre de notre système, en prenant les éléments de METHONTOLOGY comme référence. L'évaluation de l'ontologie sera effectuée au chapitre 3 de ce manuscrit, après avoir exposé la méthode d'enrichissement proposée et la population du module ASTRO-THEM.

2.4.1. Spécification d'ASON

Le but de notre travail est d'utiliser ASON comme base de raisonnement à l'intérieur d'un logiciel de composition automatique de workflows basé sur la description de cas d'utilisation scientifiques. Par

conséquent, ASON doit non seulement décrire des services, mais également les intégrer dans un environnement plus global dans lequel les descriptions issues de la connaissance du domaine d'application peuvent être comprises. Sa structure doit permettre de récupérer un ensemble de services en fonction de leurs capacités et être capable de fournir une description fine des données et du contexte général d'observation (caractéristiques de l'instrument ...). Elle doit indiquer, pour la recherche demandée par l'utilisateur, quels sont les services disponibles.
Cette sélection de services doit pouvoir s'accompagner d'une composition automatique (décrite au chapitre 4) en vue de résoudre le problème posé par l'utilisateur en utilisant un enchaînement de services adapté. Pour rendre cette composition possible, tous les paramètres d'entrée et de sortie des services doivent être réconciliés sémantiquement.
Enfin, l'automatisation de la mise en œuvre des compositions produites doit être assurée. Cela implique de pouvoir représenter, outre les connaissances disponibles et les profils généraux des services, les détails techniques de leurs invocations.
Le Tableau 4 spécifie ASON, selon les critères indiqué par METHOTOLOGY.

Tableau 4: Document de spécification d'ASON

Domaine	Astrophysique
Nom	ASON : Astrophysical Services ONtology
Conception	Thierry Louge, Mohamed-Hedi Karray, Bernard Archimède
Développeur	Thierry Louge
Objectif	L'ontologie vise à la description de services dans le domaine de l'astrophysique. Ces services ne seront pas uniquement des services Web, mais également des logiciels scriptables et des bibliothèques locales par exemple. Les exigences fonctionnelles des services devront être exprimées. Pour assurer la satisfaction des exigences fonctionnelles, la description technique de ces services devra être mise en relation avec des concepts issus de leur domaine d'application.
Degré de formalité	Semi-formel, annotations et concepts hérités de langage naturel.
Cadre	ASON sera utilisée dans le cadre d'une composition automatique et sémantique de services en astrophysique.

Sources de connaissances	Ontologie HELIO, thesaurus de l'IAU, descriptions des UCDs des services de l'IVOA.

2.4.2. Conceptualisation du module générique pour les services

La conceptualisation d'une ontologie, suivant la méthode METHONTOLOGY vise à aboutir au modèle conceptuel. Il est nécessaire dans cette phase de définir le vocabulaire utilisé (les termes), et les relations hiérarchiques de ces termes. C'est également dans cette phase que sont définies les relations et le rôle qu'elles jouent dans l'ontologie.

Pour ASON, la conceptualisation s'articule autour de trois questions :

- Comment représenter des services aux profils inhabituels et hétérogènes, décrits plus haut ?
- Comment articuler la représentation des services avec la représentation de leur domaine d'application ?
- Comment obtenir une représentation sémantique du domaine d'application concerné lorsque celle-ci n'existe pas, ou seulement de façon partielle ?

Ce chapitre présente les réponses apportées aux deux premières questions, qui concernent le module GEOS. La troisième question concerne le module thématique, elle est traitée dans le chapitre suivant de ce manuscrit. Avant d'entrer dans les détails au chapitre suivant, nous pouvons indiquer brièvement que cette représentation sémantique du domaine d'application a bénéficié à la fois des représentations ontologiques existantes (HELIO, ontologies de types d'objets et de sujets de services) et des connaissances du domaine contenues dans la sémantique des représentations de l'IVOA (en particulier des UCDs) et du thésaurus de l'*International Astrophysical Union* (IAU). Ces sources ont été nos principaux points pour l'acquisition de connaissances structurées dans le domaine.

A) EXPRESSION DES SPECIFICITES DES SERVICES DU DOMAINE

Nous allons voir dans cette section que la description de certains types de services ne rentre pas forcément naturellement dans les concepts proposés par OWL-S. Notre exemple astrophysique est basé sur des services à la fois plus simples (parce que les services que nous avons à traiter sont des processus atomiques) et plus complexes que les cas d'utilisation industriels envisagés dans les ontologies de services existantes.
Nous verrons également que cette complexité peut venir du profil du domaine d'applications. Ainsi, les domaines où il existe de nombreuses façons d'exprimer des données et où l'analyse est parfois très dépendante de «paramètres non fonctionnels» qui sont en informatique souvent liés au concept de qualité de service (« *Quality of Service* », ou QoS) peuvent se montrer problématiques. L'astrophysique est un des domaines, la conceptualisation du module de services proposera donc des mécanismes permettant de tenir compte de ces spécificités.

Quel que soit le domaine, le même point de départ pour la modélisation des services est que chaque service possède des entrées et des sorties. Ces entrées et sorties sont des informations avec leurs propres unités et leurs propres formats qui font partie de la description du domaine d'applications. Par exemple, un spectre peut être donné sous la forme d'un fichier ascii ou d'un fichier FITS (*Flexible Image Transport System*) ou d'autres formats. Il peut être exprimé en erg / cm 2, Jy entre autres.

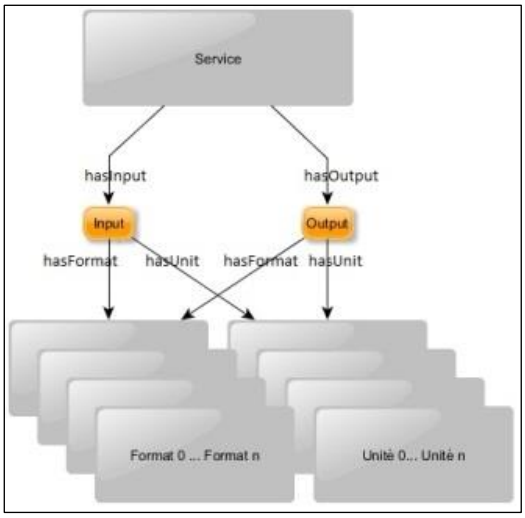


Figure 14: La multiplicité des unités et des formats

Chaque service accepte donc ses entrées et exprime ses sorties selon un ensemble variable de formats et d'unités. La Figure 14 illustre cette situation.

L'exigence de ce type de descriptions entre difficilement dans le cadre natif de OWL-S, dans lequel les entrées et les sorties d'un service sont des sous-classes d'une classe « *Parameter* » et sont assimilables à des variables qui ont un nom et un type. Chaque paramètre a un Type, qui est la classe à laquelle appartiennent les valeurs que peut prendre le paramètre. Le Tableau 5 donne la définition des entrées et des sorties de services, selon la documentation OWL-S, la Figure 15 représente graphiquement cette organisation.

Tableau 5: Définition des entrées et sorties dans OWL-S

```
<owl:Class rdf:about=«#Parameter»>
  <rdfs:subClassOf rdf:resource=«&swrl;#Variable»/>
</owl:Class>
<owl:DatatypeProperty rdf:ID=«parameterType»>
  <rdfs:domain rdf:resource=«#Parameter»/>
  <rdfs:range rdf:resource=«&xsd;anyURI»/>
</owl:DatatypeProperty>

<owl:Class rdf:ID=«Parameter»>
 <rdfs:subClassOf>
   <owl:Restriction>
     <owl:onProperty rdf:resource=«#parameterType» />
     <owl:minCardinality rdf:datatype=«&xsd;#nonNegativeInteger»>
           1</owl:minCardinality>
   </owl:Restriction>
 </rdfs:subClassOf>
```

```
</owl:Class>

<owl:Class rdf:ID=«Input»>
  <rdfs:subClassOf rdf:resource=«#Parameter»/>
</owl:Class>

<owl:Class rdf:ID=«Output»>
  <rdfs:subClassOf rdf:resource=«#Parameter»/>
</owl:Class>
```

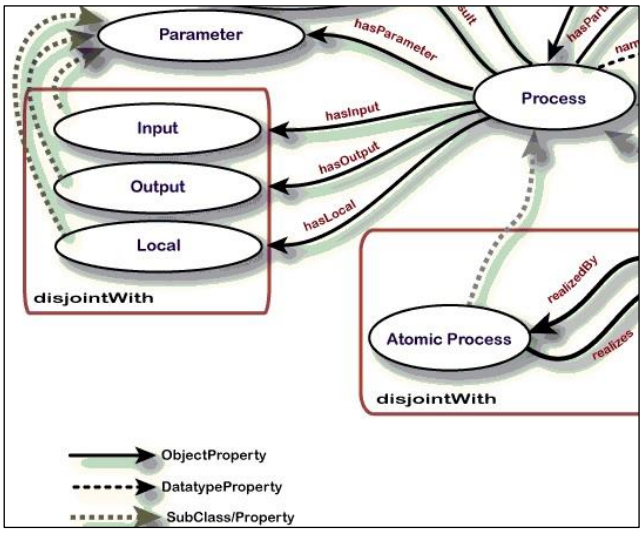


Figure 15: Les entrées et les sorties des Process dans OWL-S

Il est possible de spécifier qu'un paramètre (par exemple, l'ascension droite d'un objet, qui est une de ses coordonnées sur le ciel) a pour Type une unité particulière, par exemple des degrés décimaux. Mais ce même paramètre peut aussi avoir pour Type un angle exprimé en degrés/minutes/secondes, et il faudrait alors créer un nouveau paramètre pour indiquer un nouveau Type, ce qui ne correspond pas à la réalité. De plus, retrouver l'ascension droite indépendamment de l'unité dans laquelle elle s'exprime deviendrait alors très compliqué. Il faudrait reconnaître que les deux paramètres créés, indépendamment de leur type désignent la même quantité physique. Trouver ne serait-ce qu'une heuristique simple permettant d'opérer cette distinction est loin d'être évident. Rien n'empêche d'associer deux Types différent au même paramètre, du moins dans la définition d'OWL-S. Toutefois, cela ne résout pas le problème, puisqu'alors il faudrait pouvoir indiquer dans quel ou quel autre Type possible est exprimée une valeur donnée pour ce paramètre. De plus il nous semble que cela ne respecte pas la logique globale recherchée et qu'il reste préférable de découpler complètement le paramètre désignant la quantité physique décrite de son expression technique (format ou unité, dans notre cas).
Une autre possibilité pour utiliser les mécanismes natifs d'OWL-S permettant d'exprimer les dépendances entre entrées, sorties, formats et unités consisterait à utiliser les préconditions et les post-conditions définies dans l'ontologie. Il serait alors théoriquement possible d'indiquer qu'un service peut avoir une grande quantité de conditions préalables. Un tel service pourrait être considéré comme utilisable si un sous-ensemble donné de ces conditions était vrai (les bonnes entrées dans les bonnes unités, au bon format). Toutefois, certaines de ces conditions préalables peuvent être absolument nécessaires pour un service, alors que d'autres peuvent être facultatives pour obtenir des résultats

améliorés. Cela conduirait à exprimer des conditions multiples pour chaque processus dans OWL-S. De plus, il faudrait encore complexifier la représentation pour indiquer dans quel format ou quelle unité est effectivement exprimée une valeur concrète.

Une autre spécificité du profil des services que nous avons à traiter tient à la distinction entre les entrées obligatoires et les entrées optionnelles des services. Un service peut donner un résultat avec un noyau d'informations minimum et fournir un meilleur résultat avec des mesures complémentaires (une analyse multi-longueurs d'onde, par exemple peut résulter de l'analyse conjointe de deux à plusieurs mesures de longueur d'onde du même objet astrophysique). Comme précédemment, ces mesures peuvent être exprimées individuellement en différentes unités et formats. Cela ne signifie pas qu'un état indiquant que le service peut être utilisé doit être atteint. Cela signifie plutôt que l'invocation d'un service dépend d'un ensemble d'informations minimum qui peut être complété, comme illustré par la Figure 16.

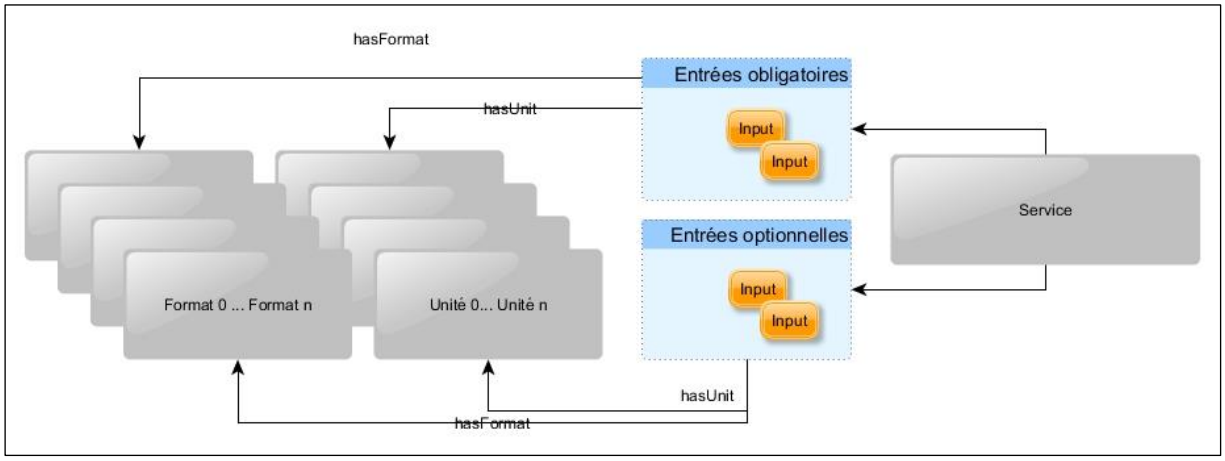


Figure 16: Entrées obligatoires et optionnelles des services

Enfin, certains paramètres d'entrée de services peuvent n'avoir de sens que s'ils proviennent d'une source unique (par exemple, une mesure et les barres d'erreur correspondantes). Certains services peuvent ainsi imposer qu'un ensemble donné d'entrées soit issu d'une source unique. Il est donc nécessaire, pour décrire complètement le profil des services que nous avons à traiter, d'exprimer quels sont les jeux de combinaisons d'informations, d'unités et de formats qui sont corrélés les uns aux autres. La Figure 17 illustre cet impératif, les liens en pointillés matérialisant les combinaisons interdites entre les sorties de services et les entrées du service suivant.

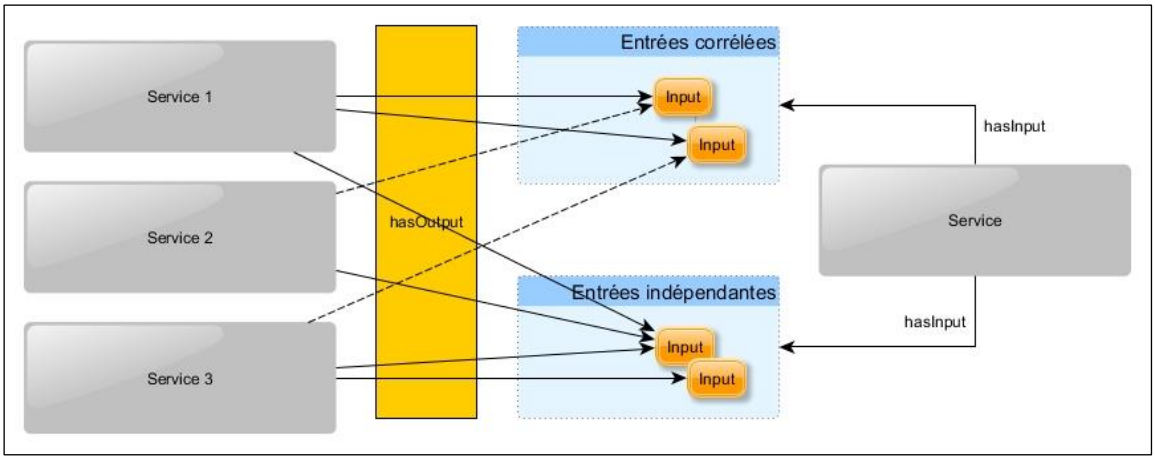


Figure 17: Entrées corrélées et indépendantes des services

Nous proposons dans la section suivante un moyen de définir les conditions que nous venons d'exposer qui est plus facilement compréhensible, plus expressif et moins verbeux en intégrant des agrégats (Severi et al. 2010) dans GEOS. Les agrégats lient les services aux informations, aux formats et aux unités de manière simple et efficace. Nous définissons des relations dans l'ontologie exprimant dans quelles unités et formats un service exprime ou accepte ses entrées et sorties. Identifier chaque combinaison de paramètres et s'assurer qu'ils peuvent être utilisés conjointement joue un rôle clé pour la sélection des services et le grand nombre de profils de services dans le domaine rend cette tâche non triviale.

B) AGGREGATION DE L'INFORMATION DANS L'ONTOLOGIE

Pour illustrer la réponse que nous apportons aux questions que nous venons de soulever, nous partons d'un profil de service **P1**, instance de **Service:Profile**, nécessitant en entrée l'information **pos.eq.ra** instance de **PosEq** exprimée dans l'unité **deg**, instance de **Unit**.

Considérons la relation:

$$HasRelevantUnit(information, unit)$$

Exprimant que l'information «information» peut être exprimée dans l'unité «unit».

La relation

$$HasInput(profile, information)$$

Indique que l'information donnée est un paramètre d'entrée pour le profil donné et la relation :

$$HasRequestUnit(profile, unit)$$

Indique que l'unité donnée est acceptable par le profil comme une unité valable pour ses paramètres d'entrée. Supposons que l'expression suivante soit vraie:

$$HasRelevantUnit(pos.eq.ra, deg) \wedge HasInput(P1, pos.eq.ra) \wedge HasRequestUnit(P1, deg)$$

Cela indiquerait qu'il est possible de donner à **P1** l'information **pos.eq.ra** exprimée en **deg** comme paramètre d'entrée acceptable. Mais ceci peut se révéler faux, parce qu'il est possible de rencontrer :

$$HasRequestUnit(P1, deg)$$

En raison du fait que **P1** accepte **deg** comme une unité valable pour d'autres informations que **pos.eq.ra** mais n'accepte **pos.eq.ra** que dans des unités différentes de **deg**. Si nous définissions un prédicat ternaire tel que

$$HasAggregation(profile, information, unit)$$

Qui indiquerait que le profil d'un service accepte l'information donnée dans une unité donnée nous pourrions régler le problème. Toutefois cela n'éviterait pas certaines confusions, par exemple:

$$HasRelevantUnit(pos.eq.ra, deg) \wedge HasInput(P1, pos.eq.ra) \wedge HasRequestUnit(P1, deg) \rightarrow HasAggregation(P1, pos.eq.ra, deg)$$

N'est pas forcément vrai dans tous les cas. Mais surtout, il n'est pas possible de définir dans OWL des relations d'arité arbitraire telles que **HasAggregation** pour exprimer quelles combinaisons de paramètres et d'unités sont définies comme acceptables pour un profil de service donné. GEOS utilise les mécanismes d'agrégations et de réconciliation pour exprimer ce besoin.

Les individus de la classe **Aggregate** lient les couples profil.information avec les unités pertinentes pour une information et un profil donnés. Les relations **HasInput** ou **HasMandatoryParam** couplées à **IsCombinedToUnit** définissent quels agrégats sont réellement utilisables par le service lié au profil, en tenant compte des unités.

Les individus de la classe **Aggregate** ont donc trois relations:

- Une au profil (soit **HasOutput**, **HasInput** ou **HasMandatoryParam**)
- Une à l'information (**IsCombinedToParam**)
- Une à l'unité (**IsCombinedToUnit**)

Un agrégat **IsCombinedToUnit** une unité et **IsCombinedToParam** une information signifie que cette information est exprimée dans cette unité. Le même mécanisme est utilisé pour les formats de données.

Sur ce modèle, nous proposons la relation **HasCorrelatedInformation**, reliant entre elles un sous-ensemble d'entrées d'un service donné, pour exprimer le cas de figure pour lequel un sous-ensemble des entrées d'un service sont corrélées. C'est encore un prédicat ternaire que nous pouvons exprimer via une agrégation.

C) SERVICES ET PROTOCOLES SPECIFIQUES

Les services IVOA sont des processus atomiques, ce qui signifie qu'un appel unique attend une réponse unique, sans dialogue supplémentaire entre le client et le serveur. Les services de l'IVOA sont très semblables à des services REST dans ce sens. Le profil général des services de l'IVOA est donc facilement identifiable, mais envoyer et recevoir concrètement des messages à destination et en provenance de ces services nécessite un mécanisme de *grounding* qui n'est pas proposé nativement par les ontologies de services existantes.

La spécification OWL-S est très claire sur le *grounding* de services WSDL, mais manque de précision sur la façon dont les services non-WSDL devraient être décrits. En outre, OWL-S ne spécifie pas comment un service est effectivement invoqué et ses résultats interprétés sans recourir à WSDL. Néanmoins, dans le cas de services accessibles via HTTP le contexte général est parfaitement connu : un service est accessible par une URL et utilise soit le protocole HTTP (applications REST) soit un protocole spécifique à la DCI préexistante du domaine, comme le protocole SSAP (Simple Spectrum Access Protocol) ou Conesearch pour l'IVOA. Nous proposons l'utilisation d'un concept « Protocol » pour refléter ce fonctionnement dans le module « GEOS » de l'ontologie de services et fédérer le mécanisme de *grounding* d'une manière simple et compréhensible. Les protocoles comportent certains paramètres d'entrée et supposent une méthode spécifique pour former le message d'envoi en vue d'effectuer les requêtes aux services. Le protocole est indépendant de l'URL d'un service, mais la façon d'exprimer les paramètres de requête et les paramètres d'entrée sont communs à chaque service qui utilise un protocole donné. La Figure 18 montre un exemple de *grounding* pour un service IVOA utilisant la description OWL-S.

Nous pouvons voir que la description du *grounding* est complexe et implique de nombreux concepts différents[44]. Il y a notamment un travail de correspondance à faire au cas par cas entre les éléments nécessaires à l'expression d'un service décrit par WSDL et ceux nécessaires à un service décrit par un autre formalisme.
Même dans le cas où la description d'origine du service utilise WSDL, les concepts « InputMessageMap » et « OutputMessageMap » peuvent être difficiles à exprimer. Ces concepts ont le rôle de décrire les messages d'entrée et de sortie des services. Ils peuvent nécessiter une transformation de type « eXtensible Stylesheet Language Transformations » (XSLT) pour assurer la concordance entre les éléments décrits dans OWL-S et les éléments issus de la description native des services. Un script permettant d'opérer cette transformation doit être fourni dans les cas où cette concordance n'est pas directe, en utilisant la propriété « xsltTransformation » de OWL-S. Dans le cas où le langage de description original du service n'est pas WSDL, la tâche consistant à établir

[44] https://www.w3.org/Submission/OWL-S/#6

manuellement ces correspondances et de les décrire convenablement dans OWL-S revient à l'utilisateur.

Des difficultés du même ordre, bien que moins importantes, apparaissent pour le renseignement des éléments « Operation » et « ParameterType » par exemple. Ces problèmes doivent être résolus au cas par cas, pour chacun des services à décrire.

```
<grounding:Grounding rdf:ID="Grounding_ix_30">
    <service:supportedBy rdf:resource="#ix_30"/>
    <!-- Groundings specifications -->
    <grounding:hasAtomicProcessGrounding rdf:resource="#Grounding_RetrieveData_ix30"/>
</grounding:Grounding>

<grounding:AtomicProcessGrounding rdf:ID="Grounding_RetrieveData_ix30">
     <!-- Reference to the corresponding operation -->
    <grounding:Operation rdf:resource="#RetrieveData_ix30_operation"/>
    <grounding:owlsProcess rdf:resource="#RetrieveData_ix30"/>
    <!-- Reference to the input message -->
    <grounding:InputMessage rdf:datatype="&xsd;#anyURI">#RetrieveData_ix30_Input</grounding:InputMessage>
    <!-- Definition of parts of input message-->
    <grounding:Input>
      <grounding:InputMessageMap>
        Specifying how the non-WSDL/SOAP mechanism is used to form the service query
      </grounding:InputMessageMap>
    </grounding:Input>
    <!-- Mapping of outputs to message parts -->
    <grounding:Output>
      <grounding:OutputMessageMap>
        Specifying how the non-WSDL/SOAP mechanism is used to understand the service results
      </grounding:OutputMessageMap>
    </grounding:Output>
</grounding:AtomicProcessGrounding>

<grounding:OperationRef rdf:ID="RetrieveData_ix30_operation">
    <rdfs:comment>
        A pointer to the operation used for retrieving data
    </rdfs:comment>
    <!-- locate port type to be used -->
    <grounding:portType rdf:datatype="&xsd;#anyURI">
      ASON.owl#http://vizier.u-strasbg.fr/viz-bin/votable/-A?-out.all+-source=IX%2F30%2Fseqp+
    </grounding:portType>
    <!-- locate operation to be used -->
    <grounding:operation rdf:datatype="&xsd;#anyURI">
      ASON.owl#http://vizier.u-strasbg.fr/viz-bin/votable/-A?-out.all+-source=IX%2F30%2Fseqp+
    </grounding:operation>
    For Non-WSDL/SOAP atomic services description in OWL-S,
        we would have to figure out if port/operation distinction is still relevant.
  </grounding:WsdlOperationRef>

<process:AtomicProcess rdf:ID="RetrieveData_ix30">
    <rdfs:label>Retrieve some data (ATOMIC)</rdfs:label>
    <rdfs:comment>Retrieves data from ix_30 service</rdfs:comment>
    <process:hasInput>
      <process:Input rdf:ID="RetrieveData_ix30_AvailableCoords">
        <process:parameterType rdf:datatype="&xsd;#anyURI">&concepts;#pos.eq.ra</process:parameterType>
        <process:parameterType rdf:datatype="&xsd;#anyURI">&concepts;#pos.eq.dec</process:parameterType>
        ... concepts / units combination has still to be expressed
      </process:Input>
    </process:hasInput>
    <process:hasOutput>
      <process:Output rdf:ID="RetrieveData_ix30_AvailableData">
       ... Insert every output concept/unit combination for service ix_30
      </process:Output>
    </process:hasOutput>
</process:AtomicProcess>
```

Figure 18: Extrait de grouding de ix_30, un service de l'IVOA suivant OWL-S

Dans GEOS, nous proposons un *grounding* spécifique pour chaque protocole, accompagné d'un lien entre le service et le protocole. Cela facilite à la fois la définition du *grounding* des services liés aux protocoles et la lisibilité de l'ontologie en évitant de dupliquer les paramètres d'entrée pour chacun des services individuels. Pour lier un service et son protocole, nous exprimons l'URL du service dans une classe (concept « accessUrl »). Les concepts « accessURL » et « Protocol » permettent donc de décrire un protocole, et l'adresse à laquelle un service peut être contacté. Mais pour pouvoir utiliser concrètement les services décrits dans l'ontologie, il est nécessaire de pouvoir lier le service théorique (avec ses paramètres d'entrée et de sortie, son protocole et son URL) avec une invocation pratique. Des

composants logiciels sont alors indispensables pour mettre en forme la requête, l'envoyer, puis recevoir le message de retour et enfin interpréter ce message. Ces logiciels peuvent être très génériques (par exemple pour appeler un protocole spécifique comme SSAP d'IVOA), ou bien aussi spécialisés que nécessaire (par exemple pour appeler une bibliothèque spécifique) ou quelque part entre les deux (par exemple pour appeler un service REST, ce qui est générique mais avec des paramètres d'entrée spécifiques au service). « QuerySoftware » est la classe qui documente ces morceaux de logiciel dans l'ontologie. Exprimer le *grounding* par des protocoles et des composants logiciels permet de séparer les protocoles des services, et de généraliser ainsi le *grounding*. Il est également plus facile d'exprimer les cas de *grounding* où http n'est pas le mécanisme de transport ; comme pour des bibliothèques locales, des dépôts de données (par exemple pour accéder à des bases de données) qui peuvent avec ce système être décrits dans l'ontologie comme tout autre service indépendamment de la couche de transport. Pour pouvoir exprimer ce mécanisme, nous devons définir le *grounding* des protocoles utilisés par les services, comme le montre la Figure 19.

```
<ObjectPropertyAssertion>
        <ObjectProperty IRI="#HasMandatoryParam"/>
        <NamedIndividual IRI="#cs:ConeSearch"/>
        <NamedIndividual IRI="#deg_pos.eq.dec"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
        <ObjectProperty IRI="#HasMandatoryParam"/>
        <NamedIndividual IRI="#cs:ConeSearch"/>
        <NamedIndividual IRI="#deg_pos.eq.ra"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
        <ObjectProperty IRI="#HasMandatoryParam"/>
        <NamedIndividual IRI="#cs:ConeSearch"/>
        <NamedIndividual IRI="#deg_radius"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
        <ObjectProperty IRI="#HasQuerySoftware"/>
        <NamedIndividual IRI="#cs:ConeSearch"/>
        <NamedIndividual IRI="#ConeSearchQuery.py"/>
</ObjectPropertyAssertion>
```

Figure 19:*Grounding* d'un protocole dans ASON

La Figure 20 illustre le *grounding* final d'un service basé sur un protocole défini dans ASON.

```
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#UsesProtocol"/>
    <NamedIndividual IRI="#ix_30"/>
    <NamedIndividual IRI="#cs:ConeSearch"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#isAccessedThrough"/>
    <NamedIndividual IRI="#ix_30"/>
    <NamedIndividual IRI="#http://vizier.u-strasbg.fr/viz-bin/votable/-A?-out.all+-source=IX%2F30%2Fseqp+"/>
</ObjectPropertyAssertion>
```

Figure 20:*Grounding* de ix30, un service d'ASON servi par le protocole cs:ConeSearch

L'utilisation d'un protocole n'est pas obligatoire, et ne convient pas à tous les types de services. Par conséquent, il est possible pour un service de spécifier uniquement le logiciel de requête et les paramètres d'entrée compris par ce dernier. Toutefois quand un protocole est utilisé, il est nécessaire de fournir toutes les informations d'entrée requises par ledit protocole. En d'autres termes, les paramètres d'entrée du protocole deviennent les paramètres d'entrée du service. ASON utilise les règles SWRL pour exprimer que, si un service peut être interrogé en utilisant un protocole donné, alors les services doivent correspondre aux exigences spécifiques du protocole.

Par exemple:

$$UsesProtocol(?s,?p), HasMandatoryParam(?p,?aggregate), Service.owlpresents(?s,?profile) \rightarrow HasMandatoryParam(?profile,?aggregate)$$

Est la règle SWRL exprimant que le paramètre «aggregate» est obligatoire (relation HasMandatoryParam) pour le service «s» utilisant le protocole p (relation UsesProtocol) représenté par le profil «profile » (relation Service.owlpresents).

D) MODELE CONCEPTUEL

La phase de conceptualisation aboutit, comme nous l'avons dit au modèle conceptuel de l'ontologie. Le modèle conceptuel d'ASON est exposé dans la figure et le tableau suivants. La Figure 21 expose les éléments principaux de la définition des services ASON, et le Tableau 6 résume les concepts du module GEOS décrits dans la section précédente

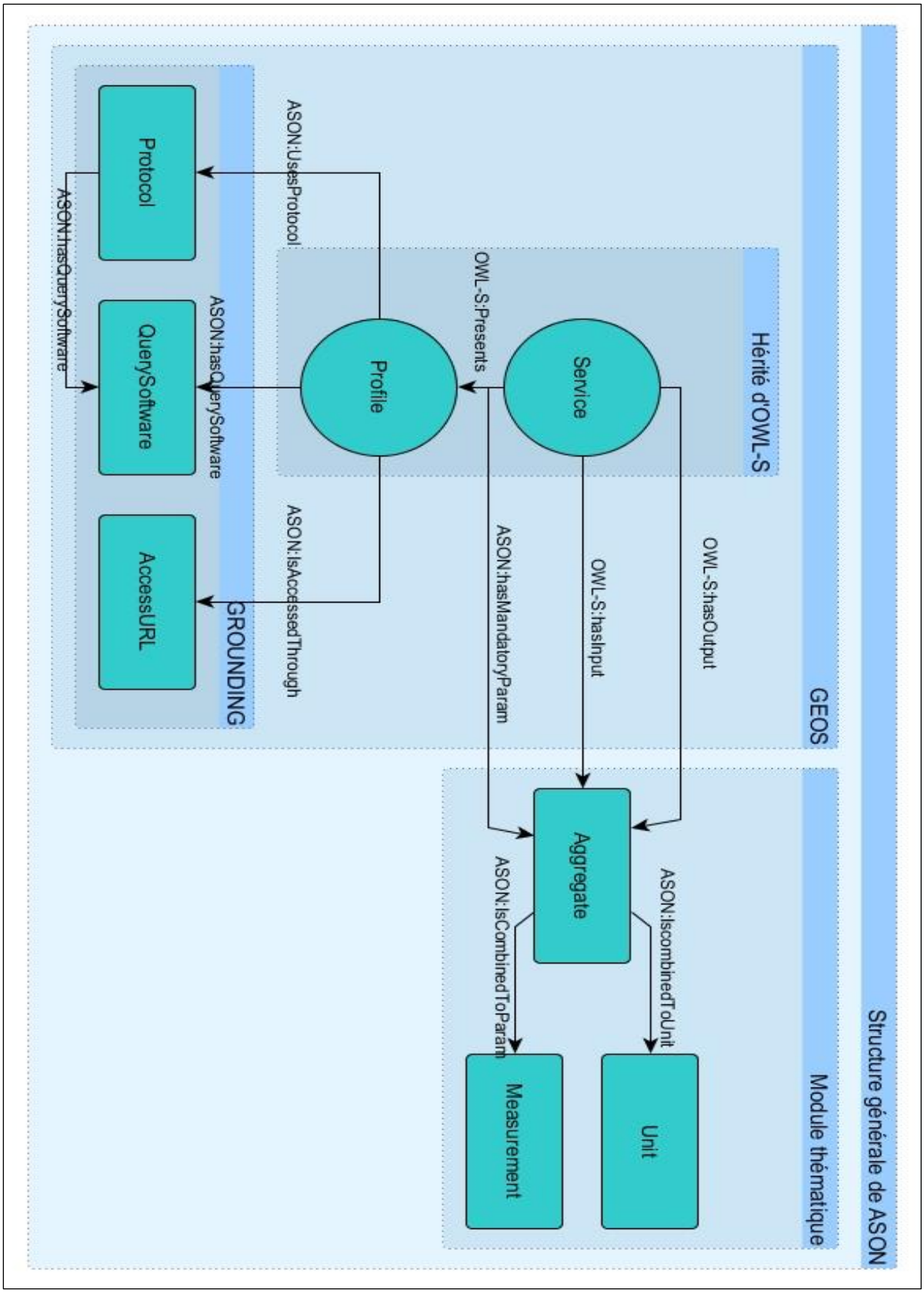


Figure 21: Structure générale d'ASON

Tableau 6: Résumé de la structure de GEOS

Individu	Documentation
Unit	Une unité (Jy, km/s…)
Measurement	Description d'une quantité astrophysique (magnetic field measurement, name of a target…) qu'un service peut fournir (sortie) ou dont il a besoin pour être invoqué (entrée).
Aggregate	Assemble une Unit et un Measurement, et lié à un service. Exprime que la combinaison Unit/Measurement combination est disponible en entrée ou sortie d'un service.
Protocol	Le protocole utilisé pour interroger un service. Facultatif.
QuerySoftware	Un logiciel local permettant de gérer les requêtes aux services ou d'utiliser un protocole.
AccessURL	L'URL d'un service (pour les services Web).

2.4.3. Formalisation et implémentation du module générique pour les services

L'approche retenue pour la description de services, basée sur OWL-S est indépendante du domaine d'applications. Toutefois, les services doivent faire référence à des éléments sémantiques issus de ce domaine d'applications, et bien que séparés les deux aspects sont nécessaires pour parvenir à la description complète d'un service et sont en relation étroite. En conséquence, nous avons conçu ASON comme une ontologie modulaire (Abb et al. 2012). Un module d'ontologie est une sous-partie autonome d'une ontologie plus vaste, entretenant des relations précises avec les autres modules de cette ontologie. Les modules sont susceptibles d'être réutilisés dans d'autres contextes (Doran 2005), et il est important dans leur définition de fixer leurs limites respectives.

Dans notre approche, la description des services sera contenue dans un module entièrement dédié à ce rôle. En conséquence, ASON fournit un «module noyau», GEOS (GEneric Ontology for Services) qui fournit la structure contenant les concepts de base utilisés pour décrire les services. Ce noyau est accompagné d'un « module thématique » décrivant la connaissance du domaine d'applications. Nous avons ainsi la possibilité de fournir une description de services indépendants du domaine et un module spécifique pour la description du domaine. Le développement des ontologies fait l'objet de nombreuses recherches. Par analogie avec le domaine de la conception logicielle, certaines approches issues de ces recherches visent à réutiliser des parties d'ontologies appelés «Content Ontology Design Patterns» (CPs) (Presutti & Gangemi 2008) rappelant le principe de « Design Patterns » en conception orientée objets. La structure modulaire d'ASON permet de considérer le module GEOS comme un CP, réutilisable pour décrire des profils de services rencontrant des spécificités semblables à celles décrites dans ce chapitre.

Le module GEOS hérite de certains concepts d'OWL-S, tout en apportant les ajouts décrits dans les sections précédentes. GEOS écarte également certains aspects d'OWLS-S. Par exemple, OWL-S contient la notion de précondition, définie comme un état du monde à satisfaire avant de pouvoir invoquer un service. Toujours dans la nomenclature OWL-S, l'effet d'un service désigne les changements sur l'état du monde créés par l'invocation de ce service, en fonction du résultat (succès ou échec) de son invocation. La documentation ne donne pas de directives sur les moyens à mettre en œuvre pour fédérer les descriptions de ces effets, ces résultats et ces préconditions autour d'une représentation commune lorsque plusieurs services coopèrent pour accomplir une tâche. Nous avons montré que ces considérations peuvent être superflues et ces mécanismes conditionnels d'invocation inutiles pour certains cas d'application comme celui des services de l'IVOA. Mais d'un point de vue plus général, le fait d'utiliser une ontologie modulaire contenant une représentation unifiée du domaine d'application permet toutefois de résoudre ces questions de vocabulaire commun. Le module thématique peut non seulement unifier la description des entrée et des sorties, mais également la description des conditions préalables à l'invocation et des effets de l'invocation des services lorsqu'elle s'avère nécessaire. La Figure 22 expose un extrait de l'implémentation concrète du schéma conceptuel.

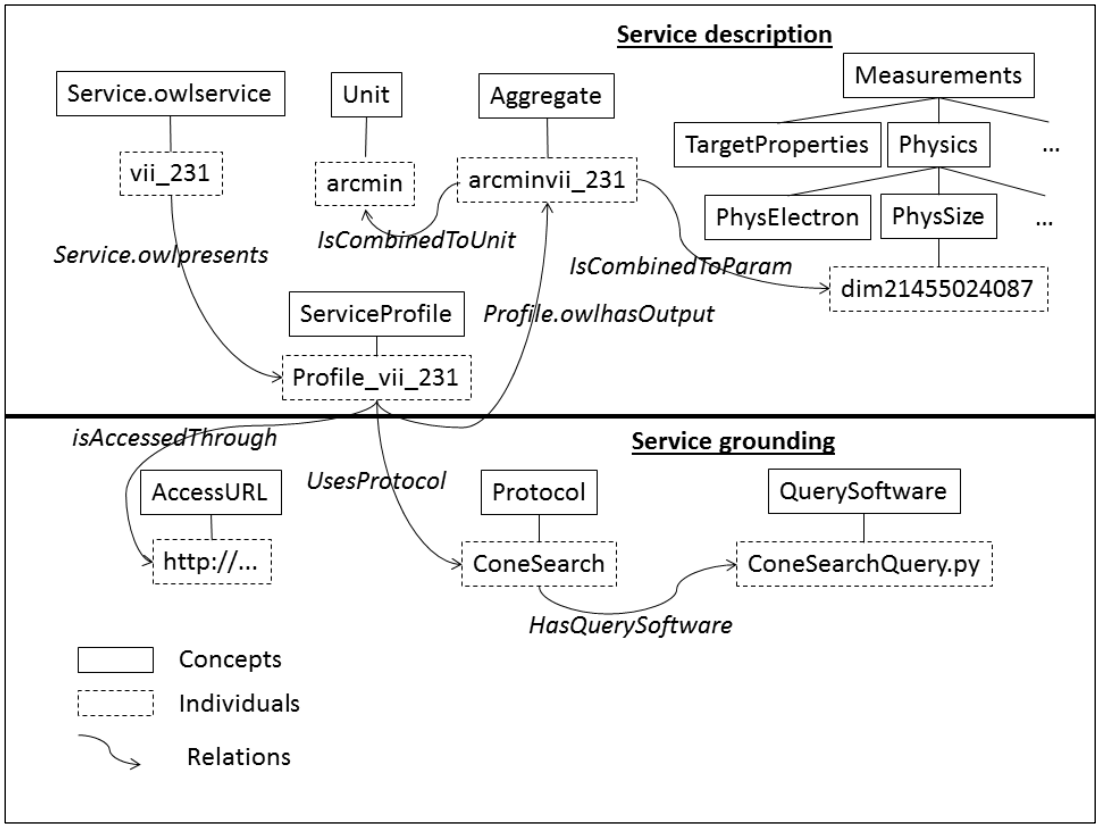


Figure 22: Implémentation de la structure d'ASON

2.4.4. Maintenance

Les futures utilisations d'ASON et la maintenance de l'ontologie doivent être prises en compte dès le début, car les futures activités de maintenance devront par exemple tenir compte de l'évolution des

modèles de description fournis par les services (lorsqu'ils existent tels que les services de l'IVOA) et à l'apparition de nouveaux protocoles. La maintenance devra donc se concentrer sur la mise à jour des mécanismes d'intégration des descriptions issues d'une DCI particulière (l'IVOA pour l'astrophysique) dans la structure de l'ontologie, malgré les modifications susceptibles d'être apportées aux protocoles, aux schémas de descriptions et aux modèles de données, par exemple. La définition de nouveaux individus ou de nouvelles classes dans l'ontologie peut apparaître (comme de nouvelles méthodes d'observation ou de nouveaux types de résultats). Néanmoins, la structure de base a été définie en fonction de la connaissance du domaine et en tenant compte des représentations existantes. Cette structure peut être étendue et rester pertinente, pour intégrer les nouveaux éléments qui peuvent apparaître dans la connaissance du domaine ; ou la formalisation des services issus de la DCI de ce domaine.

2.5. Conclusion

Le présent chapitre a proposé une structure pour un module d'ontologie qui permet de décrire des services hétérogènes par nature (bibliothèques logicielles, scripts, services Web…) et par implantation (RESTful, architecture spécifique à une DCI). Nous avons expliqué comment ce module hérite de concepts issus d'OWL-S et les complémente pour répondre aux exigences fonctionnelles particulières de son domaine d'application. Nous avons voulu exprimer ces spécificités de la façon la plus générale possible, pour que la structure soit réutilisable dans d'autres configurations impliquant des profils de services similaires, avec des exigences fonctionnelles semblables à celles rencontrées dans le cas d'application astrophysique.

L'utilisation d'ASON comme annuaire de services, ontologie globale de services et ontologie de domaine à travers le module thématique est illustrée par la Figure 23: Situation d'ASON par rapport aux approches existantesFigure 23. La partie haute de cette illustration schématise le fonctionnement de la plupart des approches actuelles. Elle se compose de l'implémentation d'une ontologie de service par service à décrire, d'un annuaire de services et d'un vocabulaire commun obtenu par l'alignement de plusieurs ontologies décrivant le domaine d'application. L'utilisation d'ASON composée de GEOS, module générique pour les services et d'un module thématique simplifie l'architecture habituelle.

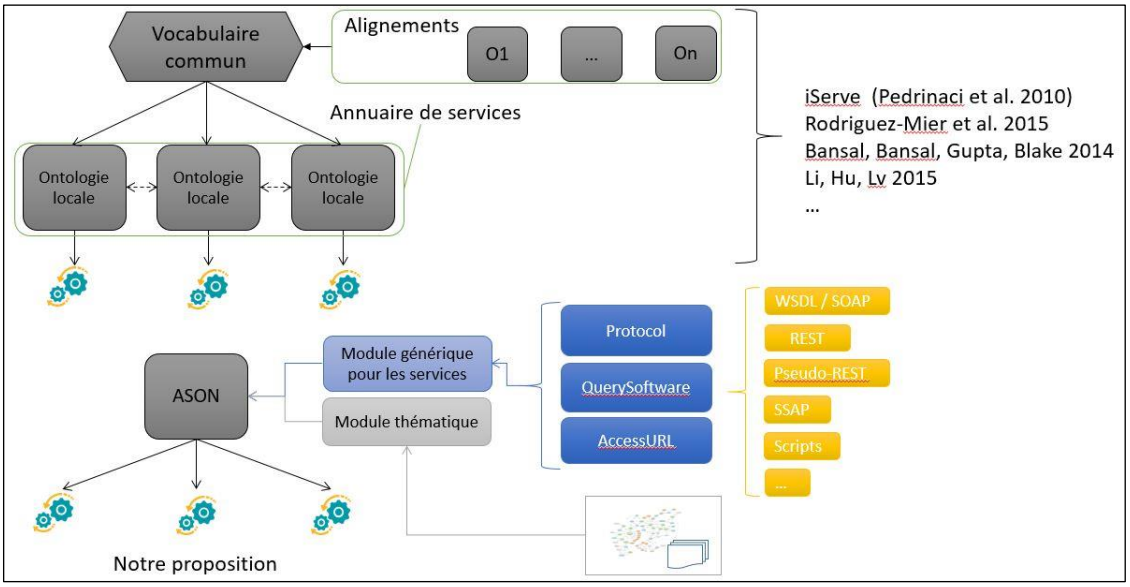


Figure 23: Situation d'ASON par rapport aux approches existantes

Nous avons vu qu'ASON se compose de deux modules, et nous avons proposé que GEOS[45] puisse être considéré comme un Content Ontology Design Patterns (CP) afin de permettre sa réutilisation dans des contextes différents que le présent cas d'application. Pour proposer un CP qui puisse effectivement jouer son rôle de brique de base pour d'autres utilisations, il existe quelques exigences à satisfaire (Presutti & Gangemi 2008) :

- La représentation du CP en OWL doit être disponible, ce qui est le cas pour GEOS
- Le CP doit être réduit (idéalement, de deux à dix classes avec des relations entre elles). GEOS ne satisfait que partiellement à ce critère, puisqu'il importe des composants de OWL-S qui font augmenter ce nombre de concepts jusqu'à 96 concepts et 91 relations. Néanmoins, les concepts de GEOS hors OWL-S ne sont qu'au nombre de 6 avec 7 relations. Il faudrait découpler GEOS des concepts de service et de profil issus de OWL-S pour satisfaire cette seconde exigence. La faisabilité serait à étudier.
- Les inférences à l'intérieur d'un CP doivent être possibles, ce qui est bien le cas ici.
- La visualisation du CP doit être compacte et intuitive, ce qui est le cas tel qu'exprimé dans les figures et le tableau ci-dessus.

[45] http://cta1.bagn.obs-mip.fr/GEOSv2.0.owl

- Les composants doivent avoir un sens linguistique, nous avons essayé de respecter cette exigence en donnant des noms explicites aux relations et aux concepts utilisés.
- La définition du CP doit provenir d'un cas réel d'application, ce qui est le cas.

Nous voyons que, bien que tous les critères ne soient pas parfaitement remplis, GEOS peut être considéré comme très proche de la définition d'un CP. Ce module permet de représenter des services Web, et plus généralement, tous les logiciels partageant un domaine d'application commun indépendamment de leurs aspects techniques. Cela aide à établir un pont entre différentes infrastructures existantes dans un même domaine d'applications (comme IVOA, HELIO pour l'astrophysique), entre des services Web indépendants et des logiciels non accessibles via Internet, comme des bibliothèques locales.

Certains éléments de GEOS proviennent de OWL-S, et d'autres répondent aux spécificités du profil des services dont traite ce manuscrit. Découpler les composants spécifiquement conçus pour permettre l'expression des profils particuliers rencontrés dans notre cas d'application des composants OWL-S peut être étudié.

ASON définit les liens entre la description des services exprimée dans GEOS et leur domaine d'application, exprimé dans le module thématique ASTRO-THEM. Ce module thématique est une ontologie à part entière, qu'il est nécessaire de construire, d'enrichir et de peupler. Le chapitre 3 traite de ces questions et des problèmes qu'elles soulèvent. Il présente les réponses que nous avons apportées, et l'illustration de la mise en œuvre de ces réponses dans le cas d'application astrophysique.

Chapitre 3: Construction semi-automatique d'un module de domaine

3.1. Introduction

Ce chapitre présente une méthode d'enrichissement et de peuplement d'un module ontologique de domaine. Ce module a pour vocation d'associer la connaissance générale du domaine d'application considéré avec les entrées et les sorties des services décrits dans le module de service. Son rôle est donc proche de celui d'une ontologie de domaine, mais son contenu est très lié à la connaissance extraite de la description des services. Le niveau d'abstraction exprimé dans ce module est donc inférieur à celui généralement employé dans une ontologie de domaine, mais en contrepartie la précision des concepts est plus importante.

Comme pour toute ontologie, l'enrichissement et la population de ce module constituent deux étapes importantes dans son cycle de vie. Après avoir présenté quelques éléments de contexte concernant la construction d'ontologies et l'état de l'art des méthodes actuelles, nous en identifierons certaines limites liées aux impératifs rencontrés en astrophysique. Nous exposerons ensuite une méthode utilisant des textes courts, non structurés, dédiés à des experts pour l'enrichissement et la population du module.

3.2. Eléments de contexte

La construction automatique et semi-automatique d'ontologies, comme leur population, s'appuie sur des techniques d'extraction d'information utilisant le traitement du langage naturel. Les techniques d'EI extraient des relations en analysant des structures de texte par NLP, et les résultats sont filtrés pour identifier des entités contenues dans le texte à l'aide d'ontologies génériques (ou spécifiques au domaine de connaissance) comme WordNet, FreeBase, BabelNet ou des ontologies privées qui servent de références de connaissances.

Ce chapitre vise une telle extraction, dans le but d'enrichir et de peupler ASTRO-THEM, le module thématique de ASON. L'appel à un service astrophysique renvoie des quantités astrophysiques. Les descriptions qui renseignent l'utilisateur d'un service sur les quantités que le service peut fournir sont appelées les capacités du service. Ces informations sont composées de texte court et grammaticalement non structuré, et utilisent un vocabulaire spécifique au domaine. Nous pouvons nous rapporter au Tableau 3, reproduit ci-dessous qui contient un exemple de ces capacités.

Tableau 7 : Extrait de définitions de services IVOA

Service 1

```
<column>
   <name>Hmag</name>
   <description>? 2MASS H magnitude (1.6um)</description>
   <unit>mag</unit>
   <ucd>phot.mag;em.IR.H</ucd>
</column>
<column>
   <name>e_Hmag</name>
   <description>? Mean error on H magnitude</description>
   <unit>mag</unit>
   <ucd>stat.error;phot.mag;em.IR.H</ucd>
</column>
<column>
   <name>Kmag</name>
   <description>? 2MASS Ks magnitude (2.2um)</description>
   <unit>mag</unit>
   <ucd>phot.mag;em.IR.K</ucd>
</column>
```

Service 2

```
<column>
   <name>Kmag</name>
   <description>? DENIS Ks-band magnitude (5)</description>
   <unit>mag</unit>
</column>
<column>
   <name>Kcorr</name>
   <description>DENIS Ks-band correlation factor</description>
</column>
<column>
   <name>Kxpos</name>
   <description>X-position in DENIS K-band image</description>
   <unit>pix</unit>
</column>
```

La description des informations contenues dans un service est incorporée dans les balises « <column> </ column> ». Les balises «Name» sont des descripteurs spécifiques au service. Les balises «Description» contiennent l'expression naturelle, donnée par les astrophysiciens, de la quantité. Les balises «Unit» affichent l'unité et la balise «ucd» fait référence à un mot dit « sémantique » défini dans le vocabulaire de l'OV. Comme le montre le tableau 2, ces UCDs peuvent ne pas être utilisés par tous les services. La description des capacités dépend des conventions utilisées (Ks-band ou simplement Ks est indiquée), inclut souvent le nom de l'instrument ou du programme scientifique dont les données

fournies sont le résultat (DENIS, 2MASS), avec une précision hétérogène (les micromètres, um ne sont indiqués que dans la description du service 1).

Les efforts conduits au sein de l'OV sont orientés vers la définition de modèles de données communs, de protocoles d'accès aux données et de logiciels compatibles OV qui utilisent ces protocoles et comprennent ces formats. Les astrophysiciens peuvent accéder aux services des fournisseurs de données par le biais de logiciels compatibles OV et utiliser les données pour leurs propres recherches. Néanmoins, la compréhension fine des données contenues dans les services astrophysiques demeure un défi. La description des modèles de données et le thésaurus existant fournissent un niveau élevé d'abstraction, alors que la définition interne contenue dans les services eux-mêmes est, au contraire, très détaillée et parfois spécifique aux instruments ou aux programmes scientifiques. De plus, l'OV n'est pas le seul moyen d'accéder à des données astrophysiques, et de nombreux instruments ou programmes scientifiques fournissent leur propre service Web, comme la base de données galactique HyperLeda (Makarov et al. 2014). Il y a plusieurs raisons à cette dichotomie, la plus souvent rencontrée étant le désir d'offrir un niveau de détail aussi précis que possible en évitant les limites imposées par les nécessités de la normalisation. Fournir à l'utilisateur un point d'accès unique pour un ensemble défini de mesures est également un argument pour de tels services. En outre, les mesures qui sont très spécifiques à certains instruments peuvent ne pas trouver de contrepartie dans les normes descriptives, ou bien ces normes peuvent imposer des conditions qui peuvent ne pas avoir de sens pour certaines données.

Dans le but de fédérer les descriptions hétérogènes évoquées précédemment, nous avons proposé que l'ontologie décrite dans ce mémoire comporte un module thématique qui joue le rôle d'une ontologie de domaine. Néanmoins plus de 12 000 services existent dans l'OV, et ceci uniquement pour le plus utilisé de ses protocoles. Décrire un tel nombre de services au sein d'une ontologie commune n'est pas une tâche facile ; le fait de faire correspondre les capacités de service avec le contenu ontologique est non trivial, prend du temps et nécessite une certaine expertise. En conséquence, plusieurs étapes de la construction de l'ontologie demandent à être automatisées. A partir d'une structure issue de l'étude des représentations existantes dans le domaine, l'enrichissement de l'ontologie déduit des descriptions des capacités des services est une de ces tâches. La population de l'ontologie, consistant à identifier les correspondances pertinentes conduisant à la bonne description d'un service et de ses capacités à l'intérieur de l'ontologie nécessite également une automatisation. La section suivante présente certaines des méthodes de construction et de population automatique d'ontologies.

3.3. Etat de l'art

La conception automatique d'ontologies a connu de grandes avancées grâce à des travaux de recherche récents, tant au niveau de son automatisation que dans les résultats obtenus. L'extraction de l'information (EI) est un domaine de recherche visant à extraire des concepts et des relations entre ces

concepts à partir de documents relatifs au domaine à décrire. Ces relations et concepts peuvent être utilisés pour la conception, l'enrichissement et la population des ontologies.

Néanmoins, une connaissance préexistante du domaine reste nécessaire pour que ces approches réussissent. Cette connaissance doit être mise en relation avec le contenu des textes dont on cherche à extraire l'information. Le processus menant à l'intégration de connaissances à l'intérieur d'une ontologie à partir d'un corpus de documents est appelé l'acquisition de connaissances de domaine (Zouaq & Nkambou 2008). Il comprend, outre les étapes de reconnaissance des informations dans le corpus, des étapes de validation de la connaissance extraite. Ces étapes de validation assurent la fiabilité de ce processus semi-automatique, qui dépend de sources de connaissances bien établies en-dehors du corpus à étudier. Ces étapes de validation peuvent être automatiques, ou semi-automatiques suivant les approches considérées.

Les auteurs de l'analyse des concepts sémantiques (SCA) présentent un exemple testé avec 310 textes de 500 à 1000 mots (Tushkanova & Gorodetsky 2015), ce qui est très différent de la longueur moyenne de 10 mots des 149056 textes rencontrés dans les descriptions des services de l'IVOA. SCA s'appuie sur DBpedia et d'autres sources pour identifier des concepts de haut niveau d'abstraction à mettre en correspondance avec les concepts identifiés dans les textes eux-mêmes. Il existe des cas où un niveau de détails important dans la séparation de concepts est nécessaire, ou qui manquent de sources de connaissance suffisamment couvrantes et exploitables. Des sources comme DBPedia peuvent alors se révéler peu informatives, ou trop générales. Cette approche n'est pas adaptée pour ce type de cas.

Les extracteurs d'information « ouverts » (Open IE) traitent du problème d'extraction de relations à partir du texte sans avoir besoin d'un ensemble de relations-types prédéfinies, issues de sources de connaissances préexistantes. Dans les travaux qui ont mené à la création de l'extracteur de relations ReVerb, les auteurs de (Fader et al., 2011) ont réussi à surmonter le besoin de connaissance de domaine en arrière-plan pour l'extraction des relations à partir de texte. Bien que ReVerb ne soit pas conçu pour extraire des taxonomies mais pour extraire des relations, il est très efficace dans de nombreuses configurations(Fader et al., 2011). ReVerb extrait les relations verbales, en supposant qu'il y ait un verbe dans la phrase, et que la structure de la phrase soit grammaticalement construite autour du verbe.

OLLIE (Mausam et al. 2012) est un Open IE qui utilise ReVerb et élargit ses possibilités. OLLIE donne des informations de contexte, classe les relations en fonction de la probabilité de l'exactitude de leur extraction, et reconnaît davantage de structures verbales. Néanmoins, OLLIE n'a identifié que 19729 relations sur les 149056 descriptions de quantités différentes disponibles dans les définitions des services de l'OV. De plus, la plupart de ces relations ont identifié des concepts non pertinents, en raison de l'absence de contenu explicatif et de l'absence de contexte propre à expliciter ces définitions.

La méthodologie exposée dans (Chen et al. 2015) est un autre exemple d'approche récente qui fait une large utilisation de dictionnaires de termes, de marquages de termes à l'intérieur des phrases,

de fréquences des termes et de similarités sémantiques. Bien que ses auteurs n'utilisent que des dictionnaires (et il existe au moins deux dictionnaires de haut niveau disponibles en astrophysique), la méthode globale repose toujours sur des typologies de phrases et de représentations sémantiques; qui peuvent justement ne pas être disponibles comme dans notre cas d'application astrophysique.

Les travaux de Cronin et al. (Cronin et al. 2011) qui ont conduit à la création de « Never-Ending Language Learning » (NELL), utilisent les algorithmes Coupled Pattern Learner (CPL) et Coupled-SEAL (C-SEAL) décrits dans leurs article précédent (Carlson et al. 2010). Ces algorithmes ne fonctionnent pas pour l'analyse de phrases non verbales, qui ont d'ailleurs été exclues du corpus sur lequel les expérimentations pour la conception de CPL ont été menées. De plus, plusieurs notations et usages spécifiques au domaine (des mots particuliers entre crochets, des notations d'unités…) seraient considérés comme des données « bruitées » pour le fonctionnement de CPL. C-SEAL utilise un jeu de relations préétablies (les «seeds» ou graines), qui sont comparées au contenu de pages Web à travers des règles d'extraction obtenues par l'utilisation de l'algorithme SEAL. Une telle méthode d'extraction suppose à la fois un grand nombre de pages Web de référence et un grand nombre de relations préétablies.

Il est possible d'extraire des relations et des concepts d'une analyse probabiliste des distributions des schémas de termes dans un corpus à l'aide d'un réseau sémantique comme ProBase (Lee et al. 2013). Le schéma proposé pour ces extractions est le même, que ce soit pour des extractions de concepts ou d'instances et est le suivant (pour des textes en anglais):

the <*a*> of (the / a/ an) <*i/c*> [is]

Dans ce schéma, a est l'attribut cible, i et c représentent soit une instance soit un concept. Quand nous testons ce schéma sur les 149056 descriptions uniques rencontrées dans les services de l'IVOA, nous le retrouvons 350 fois (0.23%). Relâcher la contrainte sur la présence du verbe augmente ce score jusqu'à 7195 fois (4.8%), ce qui est évidemment bien trop faible pour analyser le corpus que nous avons à notre disposition.

Un autre exemple de recherches récentes concernant l'extraction de connaissances est fourni par (Pia 2015), présentant un cas d'application issu de l'archéologie Italienne. L'auteur présente une méthode de formalisation du langage pour l'extraction d'informations basée sur des ontologies (Ontology-Based Information Extraction, OBIE) utilisant l'environnement de traitement du langage naturel Nooj pour l'analyse de texte. Nooj analyse le texte au moyen d'automates finis et de transducteurs finis pour la reconnaissance de structure de phrases avec l'appui de références grammaticales et de dictionnaires. La méthode de (Pia 2015) demande par conséquent une connaissance solide du domaine et des phrases grammaticalement bien formées.

Trouver des structures permettant l'identification de règles correctes entre les concepts et les relations est au cœur de (Halevy & Noy 2016) qui prend le contenu du Web comme base pour l'identification des concepts, la désambiguïsation et la sélection des règles. Les travaux présentés dans (Halevy & Noy 2016) utilisent l'immensité du flux de requêtes Web pour identifier les caractéristiques

de fréquence entre les concepts qui partagent les mêmes mots principaux pour rejeter les associations non pertinentes entre les mots. Cela n'est pas applicable dans notre cas où chaque description est correcte dans sa formulation propre; et par conséquent aucune association rencontrée ne peut être écartée. En outre, le nombre d'associations entre les termes rencontré dans plusieurs milliers d'occurrences n'a rien de commun avec la quantité de flux de requêtes issu du Web.

L'extraction de connaissance utilisée dans le système YAGO (Suchanek et al. 2007) utilise WordNet et un jeu d'heuristiques pour identifier les catégories pertinentes pour des concepts, et le classement en catégories de Wikipedia pour détecter les hypéronymes de ces concepts. Toutefois, être capable d'identifier la composante principale, les modificateurs situés avant et après l'identification de catégories dans un champ disciplinaire hautement spécialisé demanderait une implication importante de plusieurs experts du domaine.

Une étude des approches de population d'ontologies a été menée dans (Faria et al., 2013). Le processus générique décrit dans cette étude comporte trois étapes: «Identification des instances candidates», «Construction d'un classificateur» et «Classification des instances». La première phase consiste à trouver dans un corpus donné des correspondances pertinentes par rapport à une ontologie de référence. Un classificateur est ensuite généré au cours de la deuxième étape et utilisé lors de la «classification des instances» pour associer les instances candidates trouvées dans l'identification avec les classes pertinentes décrites dans l'ontologie.

Les techniques basées sur la vectorisation des mots et plus spécifiquement Word2Vec (Mikolov et al. 2013) représentent les mots comme des vecteurs. Cette représentation a fait les preuves de son efficacité pour exprimer à la fois la distance sémantique et syntaxique entre des mots. Le réseau de neurones sous-tendant la vectorisation (CBOW and Skip-gram models for (Zhao et al. 2015)) doit être entraîné sur un corpus, indépendamment de la cohérence grammaticale des éléments de ce corpus. Par conséquence, nous avons envisagé l'utilisation de ces vecteurs de mots pour réunifier les capacités des services en se basant sur la similarité de leurs contenus.

3.4. Limites des approches existantes

Des approches concernant la conception automatique et semi-automatique des ontologies qui n'ont pas été énumérées dans la section ci-dessus peuvent être trouvées (Astrakhantsev & Turdakov 2013). Cependant, ces approches ont souvent en commun plusieurs hypothèses qui peuvent ne pas convenir à tous les contextes d'application. À titre d'exemple, les approches linguistiques supposent que les textes sont composés de phrases grammaticalement cohérentes. Les approches existantes ont besoin de sources externes de connaissances pour identifier les concepts et les relations d'un corpus donné. Ces approches donnent de bons résultats lorsque de telles sources sont disponibles, fournissent suffisamment d'informations avec le bon niveau de détail pour extraire des candidats à la taxonomie et quand la structure de ces sources de connaissances externes est connue. Il peut y avoir des domaines où de telles sources sont indisponibles, ou disponibles en nombre insuffisant, ou non standardisées.

Pourtant, aucune des approches de construction automatique d'ontologies énumérées dans l'état de l'art de ce chapitre n'évacue la nécessité d'une source externe de connaissances qui soit suffisamment couvrante et bien formée. Bien que plusieurs ontologies pour l'astrophysique en dehors d'ASON existent, elles ont été conçues pour des besoins spécifiques. Nous avons évoqué au chapitre 1.3.5 qu'une ontologie concernant les types d'objets astrophysiques (Derriere et al. 2010), et une ontologie générale des sujets relatifs aux services OV (Thomas 2015) ont déjà été produites. Néanmoins, selon nos connaissances il n'existe pas d'ontologie pour l'astrophysique proposant une couverture du domaine assez large pour supporter efficacement l'appariement entre les descriptions très spécifiques contenues dans les capacités des services et une connaissance plus générale du domaine.

Notre but est différent des travaux antérieurs portant sur des ontologies pour évoqués précédemment. Puisqu'ASON est une ontologie de services, elle doit exprimer un ensemble de concepts et d'individus qui soient le plus proche possible des capacités de ces services. Cela impose d'être en mesure d'identifier à l'intérieur des capacités de services les points communs et les différences concernant la description des informations contenues dans ces services. Cela impose donc d'être en mesure d'entrer dans les détails du contenu de ces informations. Par exemple, nous devrions être capables de séparer les coordonnées faisant référence à un équinoxe (1950 par exemple) des coordonnées faisant référence à un autre équinoxe (j2000 par exemple). Ne pas faire ce type de distinction pourrait conduire à des utilisations de services insatisfaisantes, en passant ou en récupérant des coordonnées erronées. A l'inverse, séparer trop finement les informations sémantiquement proches les uns des autres peut conduire à une sur-précision dans la conception de l'ontologie. Dans notre exemple, l'ontologie devrait être en mesure d'exprimer que les deux valeurs sont toutes les deux des coordonnées, qui ne peuvent être récupérées indifféremment que dans un contexte où une coordonnée peut être dé-corrélée de l'équinoxe. En résumé nous visons à créer l'ontologie à partir des capacités des services, en exprimant leur diversité sans perdre leur généricité. Il est nécessaire pour cela de pouvoir détecter, dans les descriptions des capacités des services, quels sont les éléments communs constitutifs d'une information précise.

Un des moyens pour retrouver des éléments communs porteurs de sens dans des descriptions hétérogènes, est d'utiliser le concept d' « unité linguistique atomique » (*Atomic Linguistic Unit*, ALU) que (Pia 2015) a adopté de (Silberztein 1993), et qui est défini comme une «unité non analysable de la langue». Une ALU peut être un seul mot ou une unité composée de plusieurs mots (par exemple « right ascension », «ascension droite» en astronomie est un terme qui ne nécessite aucune analyse plus poussée). Pour identifier les ALU dans les descriptions de services, nous utiliserons des concepts issus d'ontologies astrophysiques sur les types d'objets de l'univers (Derriere et al., 2010), de l'héliophysique (Bentley et al., 2013) et des sujets généraux des services astrophysiques (Thomas 2015). Une autre source d'intérêt pour l'identification des ALUs est la définition des UCDs de l'IVOA[46] et leur

[46] http://www.ivoa.net/documents/latest/UCDlist.html

taxonomie, conjointement au thesaurus UAT[47] qui couvre la taxonomie du domaine astrophysique à un haut niveau d'abstraction.

La première approche suivie au cours de cette thèse était basée sur un vecteur de comparaisons syntaxiques construit à partir de mesures de similarité suggérées dans (Li et al. 2006) pour la population d'ontologies. A l'aide d'un algorithme d'apprentissage automatique, chaque description candidate se voyait affectée à l'une ou l'autre des classes du «training set». Cela correspondait à l'étape de construction d'un classificateur de (Faria et al. 2013). L'approche de (Li et al. 2006) avait été adaptée pour utiliser un vecteur de comparaison syntaxique à la place du vecteur sémantique original, en raison du faible apport de connaissances sémantiques disponibles dans le domaine. Toutefois, cette approche a été limitée par le grand nombre de descriptions utilisant les mêmes termes dans des contextes très différents, et cette difficulté est grandement diminuée par l'usage de *clustering* (apprentissage non supervisé) à partir de matrices de similarité qui constitue l'approche présentée dans ce chapitre. De plus, l'apprentissage supervisé permet de positionner des éléments nouveaux dans des classes déjà définies. L'approche de la population de l'ontologie par ce type d'apprentissage peut être étudiée à partir d'un « *training set* » contenant autant de classes que possible, avec suffisamment d'exemples de descriptions appartenant à chacune de ces classes. L'apprentissage supervisé peut être une piste à explorer pour la population de l'ontologie. Toutefois, puisqu'il suppose une connaissance des résultats possibles pour le problème posé (quel élément de l'ontologie se rapproche-t-il le plus de la description considérée) il ne convient pas pour son enrichissement, qui vise au contraire à découvrir de nouvelles solutions, sous la forme de nouveaux concepts ou de nouveaux individus, au problème de la population.

La comparaison entre des ontologies existantes peut servir à obtenir une structure capable de décrire les connaissances en astrophysique. Cependant, l'utilisation de cette méthode nécessiterait une couverture ontologique suffisante des connaissances du domaine d'application. En outre, parmi les approches d'appariement d'ontologies disponibles (Shvaiko & Euzenat 2013) la meilleure façon de procéder serait d'établir des correspondances collaboratives. Cela nécessiterait d'impliquer de nombreux chercheurs pour chaque sous-champ spécifique de la discipline aussi bien qu'une deuxième étape d'enrichissement même si une structure commune était obtenue.

Nous allons exposer dans ce chapitre une méthodologie permettant :

- La définition d'une structure pour le module thématique de l'ontologie présentée dans ce mémoire
- L'enrichissement automatique de ce module thématique (ASTRO-THEM, dans le présent cas d'application) à partir de textes courts et non structurés utilisant un vocabulaire

[47] http://astrothesaurus.org/thesaurus/dendrogram/

spécifique à un domaine de connaissances. Cet enrichissement ne doit pas reposer sur une large couverture du domaine par des connaissances préexistantes.

- L'évaluation de la qualité de l'ontologie enrichie obtenue précédemment.
- La population automatique d'une ontologie avec une évaluation de la qualité de la description des services.

Pour atteindre ces objectifs, nous proposons une approche basée sur le regroupement des textes utilisés à l'intérieur de matrices de similarité, puis sur l'analyse du langage naturel sous-jacent aux sous-groupes identifiés à l'intérieur de ces matrices.

Le but de la section suivante est de présenter une méthodologie pour enrichir et peupler l'ontologie de services avec des individus issus de textes courts, non structurés et très spécialisés (la description des capacités de services). Nous avons besoin de construire une structure qui puisse être comparée avec le contenu de ces capacités pour insérer correctement la description de chaque service à l'intérieur de l'ontologie. Nous présentons donc notre méthodologie pour obtenir une telle structure et la peupler avec les individus et les concepts issus des capacités de services, rendant ainsi ces services sélectionnables et utilisables à partir de l'ontologie. La section suivante aborde l'acquisition des connaissances effectuée pour structurer ASON.

3.5. Méthode proposée

Le module thématique est issu de l'élaboration manuelle d'une première architecture, enrichie par la suite pour décrire le mieux possible le domaine d'application des services. La première architecture est obtenue à partir des connaissances préexistantes du domaine, même si ces dernières ne couvrent pas tout le domaine en question, et quel que soit le niveau de détail qu'elles indiquent. Les étapes suivantes d'enrichissement et de population ont pour rôle de compléter la couverture et d'adapter le niveau de détail pour obtenir une structure adaptée aux besoins de description des services.

Pour assurer de façon satisfaisante l'enrichissement et la population de l'ontologie, nous devons:

- Identifier les concepts du module thématique (ASTRO-THEM) qui apparaissent dans les capacités des services
- Raffiner le niveau de détail des concepts existants avec de nouveaux sous-concepts extraits des capacités de services de l'IVOA
- Compléter la structure de l'ontologie complète avec de nouveaux concepts extraits de ces mêmes capacités
- Identifier les individus qui sont des instances des concepts
- Relier les capacités des services aux concepts et aux individus pertinents

Les services de l'IVOA nous fournissent 296568 capacités. Ces capacités doivent être décrites à l'intérieur d'ASON, comme individus identifiés sous les bons concepts. Parmi les 296568 descriptions,

149056 sont différentes les unes des autres. La difficulté consiste à rassembler ces informations autour de concepts et d'individus communs, malgré l'absence de règles de description originales.
Nous proposons, pour l'enrichissement et la population du module, une approche basée sur le regroupement des descriptions à l'intérieur de matrices de similarité. Ces matrices permettent la détection de clusters de descriptions les plus similaires, qui rend à son tour possible l'analyse du langage naturel du contenu des clusters résultants. Notre travail partage certains concepts avec l'approche Evans dans le cadre NERO (Evans 2003). L'utilisation de l'apprentissage non supervisé, le regroupement en « clusters » et l'utilisation d'hypéronymes pour l'identification de ces clusters sont les principales similitudes. Bien que l'approche d'Evans s'appuie sur des mots commençant par une lettre capitale pour identifier les hypéronymes, cette méthode perd son intérêt lorsqu'elle est appliquée à du texte non structuré. De plus, NERO utilise des similitudes taxonomiques basées par exemple sur WordNet. Comme exposé précédemment, pour des domaines tels que l'astrophysique les sources de connaissances à même de soutenir cette mesure de similarité taxonomique peuvent se révéler peu nombreuses et peu couvrantes, ou peu précises.

La première étape de la méthode d'enrichissement et de population proposée dans ce manuscrit consiste à rassembler des descriptions similaires en clusters. Il est donc très important de décider à quel groupe une description doit appartenir. Etant donné que les profils des clusters ainsi constitués serviront d'éléments décisionnels pour extraire une taxonomie, il est nécessaire que ces profils soient le mieux définis possibles.

Les clusters de descriptions sont constitués selon deux critères:

- Les ALUs qui peuvent être trouvées dans les descriptions
- La valeur de similarité d'une description par rapport à toutes les autres descriptions du corpus

Lorsqu'aucune ALU ne peut être trouvée à l'intérieur d'une description, le seul paramètre devient la valeur de la mesure de similarité. L'utilisation d'un modèle Word2Vec (Zhao et al. 2015), entraîné à partir du thésaurus de l'UAT, de la définition des UCD de l'IVOA et des capacités des services elles-mêmes, fournit une estimation de la similarité entre les mots du corpus. L'obtention d'une valeur de similarité entre les descriptions du corpus est alors simple.
Toutefois, quelques mots très inhabituels ou des combinaisons inhabituelles de mots peuvent être rencontrés à l'intérieur des descriptions où aucune ALU n'est trouvée. Les valeurs de similarité données par Word2Vec fournissent dans ces cas des résultats moins bons que la méthode syntaxique basée sur des mesures de similarité largement utilisées. Les clusters sont constitués à partir de matrices de similarité qui contiennent les valeurs de similarité de toutes les descriptions qui partagent la même ALU. Pour les descriptions où aucune ALU n'est identifiée, les clusters sont constitués sur des matrices de similarité contenant des valeurs de similarité de descriptions non triées. Par conséquent lorsqu'aucune ALU n'est trouvée, les descriptions sont réparties au hasard dans des matrices de similarité. La taille de ces matrices devient alors un paramètre important pour l'identification des

clusters, et l'approche sera testée avec différentes tailles de matrices.

La deuxième étape consiste à isoler les éléments communs à chacun des membres de ces clusters et à observer la régularité de leur répartition et leur fréquence afin d'en extraire une taxonomie. Le traitement du langage naturel permet de dégager des motifs des descriptions présentes dans les clusters. Ces motifs sont des répétitions de termes appartenant à la même catégorie lexicale (verbe, nom, adjectif…), et à la même place dans différentes descriptions du cluster. Suivant la fréquence de la répétition de ces motifs par rapport au contenu général du cluster concerné, les éléments répétés peuvent révéler des classes ou des sous-classes sous-jacentes aux descriptions regroupées. Les éléments de ces groupes ainsi identifiés seront alors distribués dans la taxonomie résultante pour former les classes, les sous-classes et les individus de l'ontologie.
Les sections suivantes exposent l'acquisition de connaissances utilisée pour la construction de l'architecture de base du module, puis détaillent la méthode d'enrichissement et de population proposée dans ce manuscrit.

3.5.1. Acquisition de connaissances

L'acquisition des connaissances décrit le choix des sources à partir desquelles les connaissances sont extraites et la façon dont les concepts sont choisis pour élaborer la structure de base du module thématique de l'ontologie. Les descriptions des services de l'IVOA nous fournissent des informations générales sur les observations contenues dans le service (le nom de l'observatoire, le programme scientifique…) et des métadonnées spécifiques au contenu du service (longueur d'onde observée, unités exprimées…). Les principales sources utilisées pour obtenir la structure de la description du domaine de connaissance ont été l'ontologie HELIO et les descriptions des services de l'IVOA, ainsi qu'exprimé dans la Figure 25.

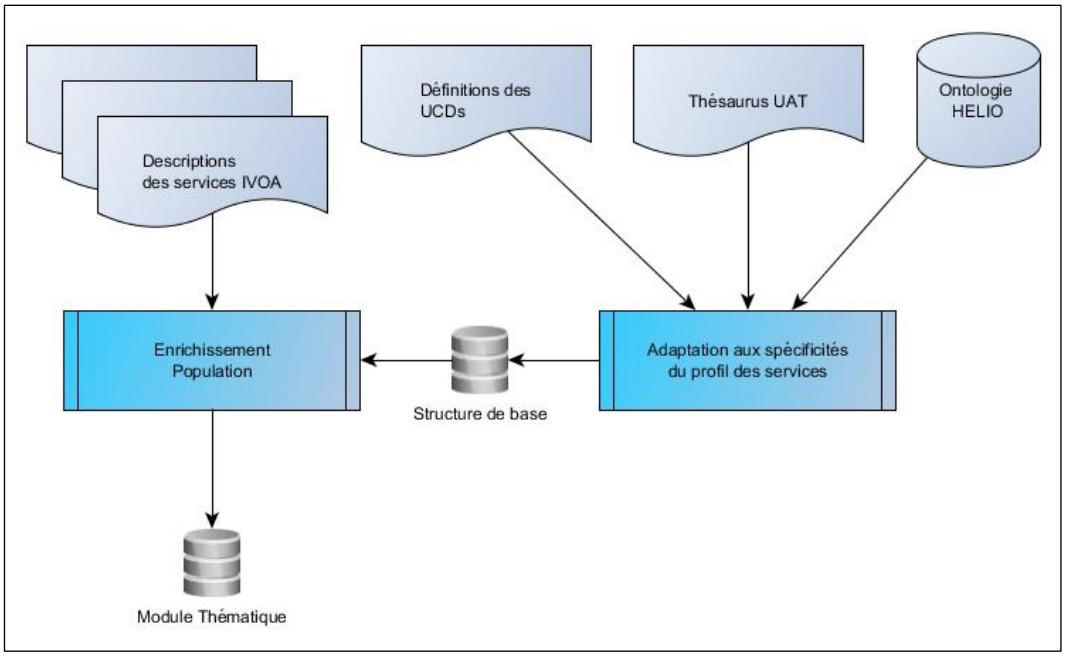


Figure 25 : Extraction d'information pour le module thématique

En plus de la description des services eux-mêmes, l'IVOA fournit une organisation hiérarchique des UCDs[48] organisés en classes et sous-classes, avec une description du vocabulaire technique compréhensible par un lecteur humain. Cette hiérarchie constitue une taxonomie annotée, dont le passage sous format ontologique est immédiat. Une source semblable, que nous avons également exploitée est un thésaurus astronomique, l'UAT(Accomazzi et al. 2014) consultable et téléchargeable en ligne[49] qui propose également une taxonomie dont les éléments sont exprimés en langage naturel.

Les concepts, et les annotations de ces concepts présents dans le module thématique de l'ontologie viennent donc des spécifications de l'IVOA, des descriptions des services et de l'ontologie HELIO. L'API REST VO-Paris[50] a été utilisée pour obtenir la description des services IVOA. Les connaissances basées sur l'IVOA et HELIO sont exprimées dans le module thématique pour décrire l'environnement des services, tandis que le module GEOS contient des concepts et des relations indépendants du domaine d'application pour la définition des services. Les services sont donc décrits complètement par l'utilisation des deux modules d'ASON. Le module thématique est construit dans une première étape par la sélection manuelle de concepts issus des ontologies existantes, et d'autres

[48] http://www.ivoa.net/documents/REC/UCD/UCDlist-20070402.html

[49] http://astrothesaurus.org/

[50] http://api.vo.obspm.fr/registry/

sources de connaissances éventuellement peu couvrantes mais permettant d'obtenir une première structure générale. Le reste de la méthodologie présentée dans cette section concerne les méthodes d'enrichissement et de population appliquées sur la structure de base obtenue pour le module thématique, en vue de sa structure définitive.

3.5.2. Abréviations, acronymes et notations

- Dx, Dy sont des descriptions d'informations données par les services (les *capacités* des services).
- $Wn(Dx)$ est le nième mot contenu dans $Dx, Wp(Dy)$ est le n-ième mot contenu dans Dy.
- $|Wn(Dx)|$ est le nombre de caractères composant $Wn(Dx)$, $|Wp(Dy)|$ est le nombre de caractères composant $Wp(Dy)$.
- $D1$ ... Dj étant des ensembles de mots W, $|Di|$ est le nombre de mots dans la description Di.
- $S(Dx, Dy)$ est la valeur de similarité entre Dx et Dy.
- E est le nombre d'éléments dans les groups de descriptions, et en conséquence le nombre de lignes et de colonnes dans la matrice utilisée pour former les clusters de descriptions.
- C est le nombre d'éléments dans un cluster.
- $AvgX_n$ est la similarité moyenne nu nième élément dans le cluster par rapport à tous les autres éléments du cluster.
- $AvgC$ est la similarité moyenne à l'intérieur d'un cluster.

$$AvgC = \frac{\sum_{n=0}^{n=C} AvgX_n}{C}$$

- $VarC$ est la variance des similarités dans un cluster.
- T est la valeur du seuil de similarité. Si $AvgC < T$, alors C est considérée inconsistante. Nous avons utilisé $T = 0.3$ pour les tests.
- NbC est le nombre de clusters inconsistants trouvés dans toutes les matrices.

3.5.3. Mesure de la similarité syntaxique

Les capacités des services sont des descriptions de l'information qu'un service est en mesure de fournir. Ces descriptions sont hétérogènes du point de vue du niveau de détail qu'elles décrivent, des conventions qu'elles utilisent et des habitudes du fournisseur de service ou du programme scientifique dont elles proviennent. Extraire une taxonomie avec ces spécificités implique de rassembler des capacités similaires (clusters de descriptions), puis diviser ces clusters en identifiant les éléments constants dans les descriptions et en trouvant la séparation appropriée entre hypéronymes (les classes) et hyponymes (sous-classes ou instances).

Les descriptions provenant des services sont soit:

- Réparties aléatoirement en groupes contenant un nombre E d'éléments, chaque élément étant une description (cas où aucune ALU n'est identifiée)

- Groupées par ALU identifiée dans les descriptions. Dans ce cas une description peur appartenir à plusieurs groupes si plus d'une ALU est identifiée dans la description concernée.

Une matrice de similarité est construite, les valeurs de la matrice étant les mesures de similarité entre les descriptions. Pour obtenir ces valeurs de similarité, la première étape consiste à ajouter à la description contenant le nombre de mots le plus bas, suffisamment de mots « neutres » pour atteindre le nombre de mots contenus dans la description ayant le nombre de mots le plus grand. Ces mots neutres assurent que la mesure de similarité est commutative entre Dx et Dy. Le choix des mots neutres est important, puisque ces mots ne doivent pas être reconnus par l'algorithme comme des candidats potentiels à un rapprochement entre les descriptions comparées, ce qui fausserait la mesure de similarité. Nous avons utilisé «###» comme mot neutre.

Les valeurs continues dans la mesure de similarité sont basées sur les mesures de similarité entre chaque paire de mots de chaque paire de descriptions, produites comme suit:

$Lev\big(Wn(Dx), Wp(Dy)\big)$ est la distance de Levensthein (Levenshtein 1966) entre Wn(Dx) et Wp(Dy).

Nous définissons $L\big(Wn(Dx), Wp(Dy)\big)$ comme la distance de Levensthein normalisée entre Wn(Dx) and Wp(Dy).

$$L\big(Wn(Dx), Wp(Dy)\big) = 1 - \left(\frac{Lev\big(Wn(Dx), Wp(Dy)\big)}{\max(|Wn(Dx)|, |Wp(Dy)|)}\right)$$

$J(Wn(Dx), Wp(Dy))$ et $JW(Wn(Dx), Wp(Dy))$ sont respectivement les distances de Jaro et Jaro-Winkler (Winkler 1999) entre Wn(Dx) et Wp(Dy).

La mesure de similarité entre $S(Wn(Dx), Wp(Dy))$ est:

$$S\big(Wn(Dx), Wp(Dy)\big) = \max\big(L\big(Wn(Dx), Wp(Dy)\big), J\big(Wn(Dx), Wp(Dy)\big), JW\big(Wn(Dx), Wp(Dy)\big)\big)$$

Nous définissons la valeur de confiance $T(Wn(Dx))$ comme le score de correspondance maximum pour un mot unique Wn de Dx envers chaque mot de Dy, après mesure des distances de Levensthein normalisée, Jaro et Jaro-Winkler:

$$T(Wn(Dx)) = \max\big(S\big(Wn(Dx), Wp(Dy)\big);\ p = 0\,..\,|Dy|\big)$$

Finalement, la valeur de similarité S(Dx,Dy) est la somme de toutes les valeurs de confiance de chacun des mots de Dx divisée par le nombre de mots |Dx| (et |Dx| = |Dy|, puisque nous avons comblé la différence de mots entre les deux chaînes par des mots neutres).

$$S(Dx, Dy) = \frac{\sum_{n=0}^{n=|Dx|} T\big(Wn(Dx)\big)}{|Dx|}$$

Ce qui précède peut être illustré par un exemple pour les deux descriptions suivantes:

$D1$ = «johnson b-v colour index» et $D2$ = «b-v (johnson) color».

Pour le mot «colour» dans $D1$ vers «color» dans $D2$, nous avons les mesures suivantes:

$$Lev(colour, color) = 1$$

$$L(colour, color) = 0.83$$

$$JW(colour, color) = 0.96$$

$$J(colour, color) = 0.94$$

$$S(colour, color) = 0.96$$

Après chaque mesure pour chaque paire de mots, nous obtenons:

$$S(D1, D2) = 0.72$$

3.5.4. Enrichissement de l'ontologie

La valeur de $S(Dx, Dy)$ entre chacune des descriptions d'un groupe sera utilisée pour former la matrice de similarité de ce groupe, sur laquelle nous appliquerons un algorithme de clustering afin de former des clusters contenant les descriptions les plus similaires entre elles. L'algorithme utilisé est celui de Frey and Dueck (Frey & Dueck 2007).

La matrice utilisée pour trouver les clusters d'informations parmi les descriptions est une matrice symétrique ($S(Dx, Dy) = S(Dy, Dx)$) composée de la façon suivante:

	D_0	...	D_x	...	D_y	...	D_n
...	1						
D_x		1		$S(Dy, Dx)$			
...			1				
D_y		$S(Dx, Dy)$		1			
...					1		
D_n						1	

L'enrichissement de l'ontologie vise à déduire des classes et des individus en analysant le contenu des descriptions. Cela améliorera à la fois le niveau de détail contenu dans l'ontologie et la précision de la population de l'ontologie au moment d'enregistrer les services. Pour accomplir cette tâche, nous appliquons l'algorithme de clustering sur toutes le matrices issues de de tous les groups de descriptions formés. L'implémentation de cet algorithme utilisée est celle disponible dans la librairie Python scikit-learn (Sofiyanti et al. 2015) dans sa version «affinity propagation». Cela fait apparaître des clusters de descriptions dans chacune des matrices, et ces clusters peuvent se révéler «consistants» ou «inconsistants».

Nous considérons que les clusters présentant une trop grande disparité des descriptions qu'ils contiennent sont peu informatifs pour l'extraction d'une taxonomie qui repose sur l'analyse des éléments communs. En conséquence, nous considérons un cluster «consistant» (c'est-à-dire dont l'analyse est pertinente pour notre propos), si $AvgC \geq T$. Dans les cas où $AvgC < T$, un cluster est considéré «inconsistant» et n'est pas analysé.

A l'intérieur de chaque cluster consistant, nous définissons l'élément le plus représentatif du cluster. Cet élément est celui qui présente le plus haut niveau de similarité moyen avec tous les autres éléments du cluster. Chacun de ces éléments est ensuite passé dans une étape supplémentaire de l'algorithme, où des groupes de E éléments représentatifs sont formés. Les similarités de ces E éléments sont mesurées, et de nouvelles matrices de similarité sont formées. De nouveaux clusters apparaissent, et ainsi de suite jusqu'à ce que l'algorithme produise un nombre d'éléments représentatifs des clusters inférieur à *E*.

Cela vaut pour les clusters consistants, mais les clusters inconsistants ne doivent pas être négligés. Comme les descriptions sont rassemblées au hasard dans les groupes (lorsqu'aucune ALU n'est retrouvée), il est possible que certaines descriptions liées les unes aux autres ne se retrouvent pas dans la même matrice. Pour surmonter cette limitation, nous réinjectons tous les éléments issus des clusters inconsistants dans les matrices de l'étape suivante. Cela permet de prendre en compte ces éléments tout au long du processus, à chaque étape.

L'algorithme générique pour la classification des descriptions, pour le cas où aucune ALU n'est trouvée est l'Algorithme 1. Lorsqu'une ALU est identifiée dans les descriptions, ces descriptions sont alors rassemblées autour de leur ALU commune et seules les étapes 2 et 3 de l'algorithme sont exécutées, le paramètre *E* étant alors le nombre de descriptions contenant l'ALU.

Algorithme 1: Clustering des descriptions sans ALU identifiée

Etape 1 Former des paquets composés d'un nombre E de descriptions
Etape 2 Former des matrices de dimensions E x E avec les mesures de similarité entre les descriptions
Etape 3 Pour chaque matrice:
    Identifier les clusters dans la matrice
    Identifier l'élément le plus représentatif de chaque cluster
Etape 4 Former des nouvelles matrices de dimensions E x E avec les éléments les plus représentatifs des clusters identifiés et chaque élément des clusters inconsistants
Etape 5 Répéter *Etape 3* et *Etape 4* jusqu'à ce que $NbC < E$

La Figure 26 fait apparaître la phase de regroupement des clusters. La réunification en clusters est l'étape qui crée les clusters finaux, illustrée à la Figure 27.

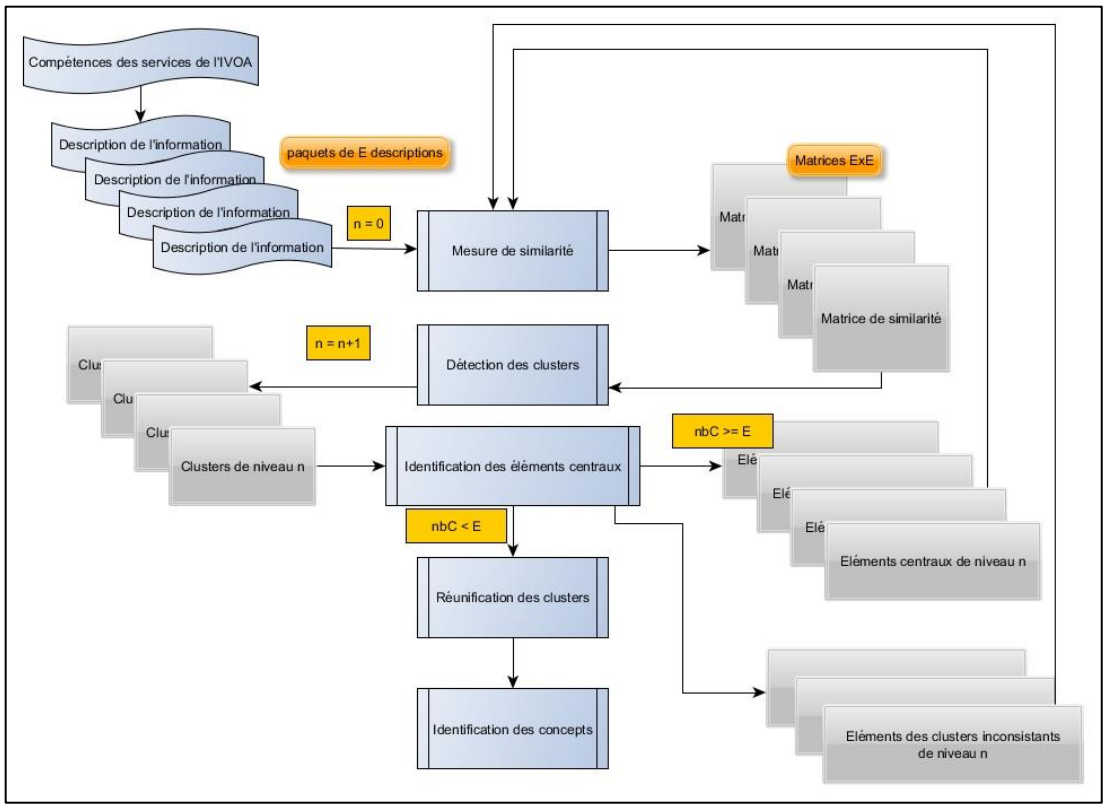


Figure 26: Méthode d'identification des concepts

La réunification consiste à prendre chaque élément de chaque cluster créé au niveau final. Chacun de ces éléments est l'élément le plus représentatif d'un groupe d'un niveau inférieur. Nous rassemblons dans le cluster final chaque élément de ce groupe de niveau inférieur, et le faisons de façon récursive jusqu'à ce que le niveau 0 des clusters soit atteint. Afin d'éviter la dispersion globale des éléments dans les clusters finaux, le tri suivant est effectué:

Pour chaque sous-niveau, chaque élément X_n dans le cluster correspondant n'est lié au cluster final que lorsque sa similarité moyenne par rapport aux autres éléments de son cluster est supérieure à la similarité moyenne du cluster moins la variance des similarités des éléments du cluster, c'est-à-dire $AvgX_n \geq AvgC - VarC$.

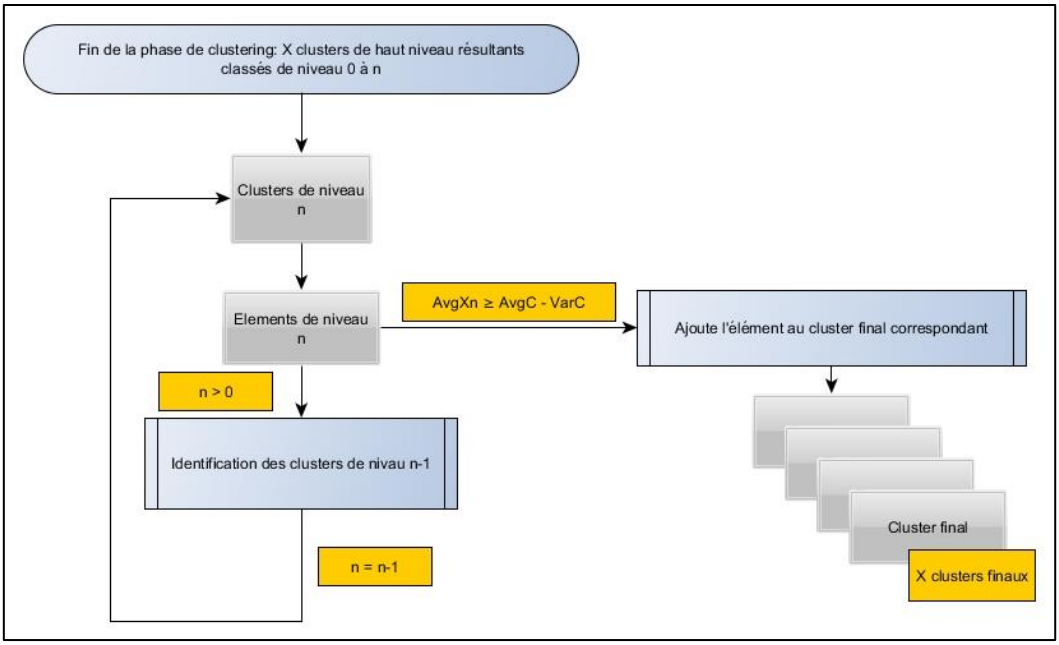


Figure 27: phase de regroupement des clusters

Cette phase de réunification forme les clusters finaux produits par notre méthodologie.

A) IDENTIFICATION DES CONCEPTS

Les clusters finaux regroupent les descriptions syntaxiquement les plus similaires, et ces clusters doivent être étudiés de plus près. Le but de l'analyse sur les clusters finaux est de déduire les classes et les individus contenus dans les descriptions qu'ils contiennent. Cela est fait en identifiant des éléments appartenant à des sources externes de connaissances (ALU) à l'intérieur du contenu de clusters, et par l'analyse du langage naturel du contenu des clusters pour trouver des sous-classes à l'intérieur du contenu de descriptions.

La première étape de l'identification des classes et des individus consiste à analyser la structure des descriptions. Cela se fait par le biais du NLP qui transforme les phrases en structures. L'étiquetage du langage naturel est un processus qui classe les mots en différents symboles, notamment en fonction du contexte de leur utilisation, et de leur catégorie (verbe, adjectif…). Une phrase se voit représentée par un motif qui exprime les différents symboles pour chaque mot qu'elle contient.

Par exemple, le NLP Part-Of-Speech (POS)-tagging produit, pour la capacité *«[14.083/24.635] sdss u band psf magnitude*» le motif suivant: «*RB CD JJ NN JJ NN NN NN*». Ce motif permet l'identification des descriptions d'un cluster qui partagent la même construction (qui n'est pas une construction grammaticale, comme l'exemple le montre).

Si une ALU se trouve à l'intérieur des descriptions, alors les motifs sont séparés en deux sous-motifs, la construction avant l'ALU et la construction après l'ALU. Si aucune ALU n'est détectée dans les descriptions, elles sont analysées dans leur ensemble. L'analyse des descriptions groupées par leurs motifs consiste à trouver la cooccurrence des termes dans un motif spécifique à l'intérieur d'un groupe.

Si la cooccurrence est assez fréquente, le terme ou groupe de termes en cours est considéré comme une spécialisation des informations contenues dans la description.

- Si une ALU a été trouvée dans les descriptions, alors nous définissons que les sous-motifs identifient une sous-classe de cette ALU.
- Si aucune ALU n'a été identifiée, alors nous posons qu'une nouvelle classe peut être créée, basé sur la cooccurrence des termes identifiés dans le motif. Cette nouvelle classe est alors placée sous la classe «Miscellaneous» dans l'ontologie, elle-même directement placée sous le contenu racine «Thing».

Un exemple du contenu d'un cluster de descriptions est donné dans le Tableau 8, la sous-classe identifiée dans ce cluster par l'analyse NLP et l'ALU trouvée dans les sources externes de connaissances définie comme classe parente pour la sous-classe identifiée. L'exemple du Tableau 8 provient d'un cluster formé dans l'ensemble de descriptions contenant l'ALU « Abundance ». Ce cluster contient 11 descriptions. L'analyse des patrons lexico-syntaxiques présents dans ce cluster, obtenus par NLP, et des termes co-occurrents révélés dans ces patrons conduit à la proposition « log number abundance » comme sous-classe de la classe liée à l'ALU correspondant, « Abundance ».

Il est possible que plusieurs sous-classes apparaissent dans un cluster. Chacune est alors nommée d'après les termes co-occurrents révélés par le NLP. Pour chaque description dans un cluster, un individu est créé qui est une instance de l'une des classes identifiées. Le choix de la classe la plus adaptée est basée sur la similarité entre la description et le nom de la classe.

Tableau 8: Descriptions à l'intérieur d'un cluster, ALU et sous-classes

Descriptions dans le cluster	**ALU (classe parente)**	**Sous-classes proposées**
log oxygen/hydrogen number abundance log nickel/hydrogen number abundance log carbon/hydrogen number abundance log magnesium/hydrogen number abundance log silicon/hydrogen number abundance log metallicity number abundance log sodium number abundance log silicon number abundance log titanium number abundance log iron number abundance log nickel number abundance	Abundance	log number abundance

B) MESURE DE LA QUALITE

La mesure de la qualité dérive directement de la méthode décrite ci-dessus. Cet indicateur de qualité peut prendre des valeurs de 0 à1, le score de 1 reflétant la confiance maximale dans le fait qu'une classe déduite du contenu d'un cluster identifie effectivement un concept pertinent dans le monde réel du cas

d'application. Pour une classe donnée, cette mesure de qualité est la similarité moyenne du cluster dont elle est déduite.

Pour une classe CL dérivant de C, la mesure de qualité $Q_{CL} = AvgC$.

Tout individu I à l'intérieur de CL est également associé à CL avec une mesure de qualité Q_I exprimant la confiance placée dans le fait que cet individu est effectivement une instance de la classe CL, et décrit un élément de la réalité du domaine. Q_I est la multiplication de la qualité de la classe par la similarité moyenne de I dans le cluster dont il est issu.

$$Q_I = AvgI \times Q_{CL}$$

Pour l'exemple du Tableau 9 ci-dessous, la qualité de la classe «log number abundance» sous-classe de «Abundance» est de 0.48, car 0.48 est le score de similarité moyenne du cluster.

Le Tableau 9 résume les mesures de qualité pour les individus appartenant à cette classe. Le Tableau 9 prolonge l'exemple du Tableau 8, la classe détectée dans le cluster est donc « log number abundance » de qualité 0.48 (la similarité moyenne des individus composant le cluster entre eux). La similarité moyenne de l'individu dans le cluster est obtenu par la formule ci-dessus, et la qualité de l'individu est le produit des deux valeurs de cette similarité moyenne de l'individu et de la qualité de la classe.

Tableau 9: Estimation de la qualité pour les individus de la classe «log number abundance»

Individu (à l'intérieur de la classe «log number abundance»)	**Qualité de la classe**	**Similarité moyenne de l'individu dans le cluster**	**Qualité**
log oxygen/hydrogen number abundance	0.48	0.5	0.24 (0.5 x 0.48)
log nickel/hydrogen number abundance	0.48	0.66	0.31 (0.66 x 0.48)
log carbon/hydrogen number abundance	0.48	0.5	0.24 (0.5 x 0.48)
log magnesium/hydrogen number abundance	0.48	0.5	0.24 (0.5 x 0.48)
log silicon/hydrogen number abundance	0.48	0.5	0.24 (0.5 x 0.48)
log metallicity number abundance	0.48	0.5	0.24 (0.5 x 0.48)
log sodium number abundance	0.48	0.5	0.24 (0.5 x 0.48)
log silicon number abundance	0.48	0.5	0.24 (0.5 x 0.48)
log titanium number abundance	0.48	0.5	0.24 (0.5 x 0.48)
log iron number abundance	0.48	0.5	0.24 (0.5 x 0.48)
log nickel number abundance	0.48	0.64	0.3 (0.64 x 0.48)

3.5.5. Population de l'ontologie

La structure obtenue par l'analyse des descriptions des capacités des services contient des classes et des individus. Elle nécessite d'être peuplée avec des services pour pouvoir être concrètement utilisée. Le mécanisme de population est beaucoup plus simple que celui d'enrichissement; puisqu'il bénéficie de tout le travail préliminaire décrit ci-dessus. Chacune des capacités de chacun des services est comparée avec :

- Tous les noms des classes, composés des termes co-occurrents trouvés dans les clusters.
- Toutes les annotations des individus de l'ontologie.

Lorsqu'une mesure de similarité retourne 1, alors la description considérée est déjà liée de façon certaine à l'individu auquel on vient de la comparer. Si aucune mesure ne retourne 1, mais qu'un maximum est atteint et que la valeur de ce maximum se situe au-delà d'un seuil minimum (que nous avons fixé à 0.5 pour les tests), alors la description a trouvé un point d'accroche dans l'ontologie. Si ce point d'accroche est trouvé en comparaison du label d'une classe, alors un nouvel individu est créé sous la classe concernée, et lié à cette description. Si le maximum est relatif à l'annotation d'un individu, alors cette description est liée à l'individu et une nouvelle annotation contenant la description est créée pour cet individu. Si le maximum est en-dessus du seuil défini, alors un nouvel individu est créé pour la description concernée sous la classe «Miscellaneous». Un extrait des résultats de la population de l'ontologie issu du cluster des Tableau 8 et Tableau 9 est exposé dans la Figure 28, tel qu'affiché dans l'éditeur d'ontologies *Protégé*[51].

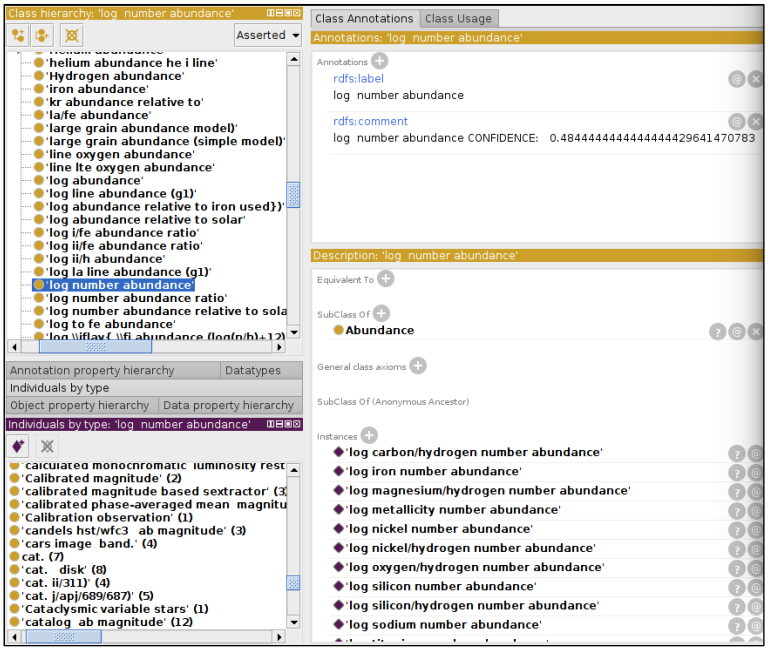


Figure 28: Extrait de la population de l'ontologie

[51] http://protege.stanford.edu/

3.6. Expérimentations, évaluation de l'ontologie

Cette section présente les résultats de l'application de la méthode exposée ci-dessus pour l'enrichissement et la population du module ASTRO-THEM. La méthode proposée a été appliquée sur 149056 descriptions de capacités de services. Parmi ces descriptions, une ALU a été identifiée dans 62250 cas et aucune ALU n'a été identifiée dans 95684 cas. La différence entre 62250+95684 = 157934 et le nombre de 149056 descriptions vient du fait que plusieurs ALUs ont été identifiées pour 8878 des 149056 descriptions.

3.6.1. Expérimentations

Nous avons vu que les classes et les sous-classes sont identifiées à partir de matrices de similarité. Les descriptions pour lesquelles aucune ALU n'est identifiée sont regroupées dans des matrices dont la taille est un paramètre important dans notre méthode. Un autre paramètre important concerne la mesure de similarité entre les descriptions elle-même, qui est obtenue à partir de la similarité entre les termes présents dans ces descriptions. Nous avons proposé une mesure de cette similarité entre les termes des descriptions, et avons signalé que Word2Vec par exemple peut aussi être utilisé pour mesurer cette similarité entre les termes. La taille des matrices utilisées et la méthode de comparaison syntaxique la plus adaptée sont donc des paramètres importants pour la méthode que nous proposons.
Pour évaluer l'influence de ces deux paramètres sur le résultat de la méthode que nous proposons, nous allons utiliser cette méthode avec différentes tailles de matrices, et avec les deux mesures de similarité. Nous nous intéresserons dans chaque cas au nombre de classes, de sous-classes et d'individus identifiés et à l'estimation de la qualité moyenne des classes et sous-classes.

Le Tableau 10 présente les résultats obtenus pour les descriptions comportant au moins une ALU reconnue. Utiliser une comparaison syntaxique basée sur un modèle Word2vec entraîné sur l'ensemble des descriptions donne des résultats dont la qualité est très légèrement meilleure que l'utilisation de la mesure de similarité proposée au paragraphe 3.5.3 de ce manuscrit. La qualité moyenne des classes, basée sur la similarité moyenne entre les descriptions continues dans les clusters dont elles sont issues, est légèrement meilleure. Plus de classes et plus d'individus sont détectés et instanciés, ce qui indique que le lien entre une description venue d'un service et un individu dans l'ontologie sera plus précis.

Méthode	Taille de matrices	Nb. de classes identifiées	Mesure de qualité des classes (moyenne)	Nb. de sous-classes identifiées	Nb. d'individus instanciés
Word2vec	N/A	2630	0.496	4972	20724
Comparaison syntaxique	N/A	1707	0.489	5885	15882

Tableau 10: Résultats obtenus par Word2Vec et comparaison syntaxique

Le Tableau 11 présente les mêmes catégories de résultats, pour les descriptions où aucune ALU n'a été détecté. En conséquence, la taille des matrices de similarité devient importante.

La méthode (Word2Vec ou syntaxique) indique la méthode utilisée pour obtenir les valeurs de similarité entre les descriptions des capacités des services. La taille des matrices indiquée dans le tableau est la taille des matrices de similarité fixées pour la détection des clusters. Le nombre de classes identifiées dans les clusters issus de ces matrices (traitées selon la méthode exposée dans la partie 3.5.4 de ce manuscrit), le nombre de sous-classes, individus et qualité moyenne des classes sont également indiquées.

Lors du traitement des descriptions dans lesquelles aucun ALU n'est retrouvé, le modèle Word2Vec donne de moins bons résultats que la comparaison proposée au paragraphe 3.5.3. Plus la taille des matrices augmente, meilleurs sont les résultats pour cette dernière mais le modèle Word2Vec a des performances moins stables que la comparaison proposée dans ce manuscrit. En effet, la qualité des classes proposées en utilisant Word2Vec diminue avec la taille des matrices mais augmente à la taille maximum testée. Dans tous les cas et quelle que soit la méthode de comparaison syntaxique adoptée, le nombre de classes diminue quand la taille des matrices augmente. Cela indique que la phase de réunification des clusters ne regroupe pas toutes les classes qui pourraient l'être, puisque le nombre de matrices avant la réunification (qui dépend de la taille individuelle de chacune des matrices) a une influence constante sur le nombre de classes identifiées.

Tableau 11: Résultats en fonction des tailles des matrices et de la méthode de mesure de similarité

Méthode	Taille des matrices	Nb. de classes identifiées	Mesure de qualité des classes (moyenne)	Nb. de sous-classes identifiées	Nb. d'individus instanciés
Word2vec	250	3638	0.336	1556	18264
Similarité syntaxique	250	9910	0.387	4410	31749
Word2vec	500	2666	0.332	1281	15105
Similarité syntaxique	500	7608	0.389	3887	27951
Word2vec	750	2341	0.317	1121	13878
Similarité syntaxique	750	6130	0.39	3370	24866
Word2vec	1000	2290	0.329	1298	14450
Similarité syntaxique	1000	5781	0.391	3384	25949

La taille des matrices et la mesure de similarité adoptée donnent des indications sur la mise en œuvre de la méthode. Toutefois, le plus important est de parvenir à qualifier la méthode elle-même indépendamment de ces paramètres, et de déterminer si elle permet effectivement d'obtenir une ontologie utilisable et fiable à partir des fragments de texte analysés.

3.6.2. Evaluation de l'ontologie

Plusieurs approches existent pour évaluer des ontologies, qui diffèrent suivant la perspective de l'évaluation (couverture du domaine décrit, qualité du schéma, efficacité des raisonnements…) et les métriques associées à cette perspective (Hlomani & Stacey 2014). Les métriques se répartissent en trois grandes catégories (Lantow & Sandkuhl 2015), (Porzel & Malaka 2004) : Les métriques structurelles, les métriques d'utilisabilité et les métriques basées sur les tâches. Les métriques structurelles s'intéressent à la structure du graphe de l'ontologie. Des mesures comme la profondeur moyenne des chemins depuis la racine jusqu'aux feuilles sont des métriques structurelles. Les métriques d'utilisabilité et basées sur les tâches s'intéressent à la lisibilité et la cohérence de l'ontologie

par rapport au domaine qu'elle décrit, selon deux axes différents. Les métriques d'utilisabilité évaluent les annotations contenues dans l'ontologie, qui sont destinées à en permettre la réutilisabilité. Les annotations étudiées sont par exemple les annotations destinées à expliciter la structure de l'ontologie, la provenance des concepts et des relations ou la version courante de l'ontologie. Les métriques basées sur les tâches visent à quantifier la cohérence de la description du domaine dans l'ontologie par rapport au monde réel. Les concepts superflus, les concepts manquants et les concepts mal identifiés (décrivant des concepts du monde réel différents de ceux qu'ils sont supposés renseigner) sont ainsi recherchés pour l'évaluation (Porzel & Malaka 2004). Quelle que soit la perspective d'évaluation choisie, identifier les bonnes métriques dans le cadre de cette évaluation est une étape qui n'est pas immédiate. Gangemi et al. (Gangemi et al. 2005) proposent ainsi 31 métriques structurelles. D'autres propositions retiennent tout ou partie de ces métriques, et définissent parfois des mesures complémentaires (Tartir et al. 2005). Suivant les approches, ces métriques sont destinées aux utilisateurs de l'ontologie évaluée, à ses concepteurs ou bien aux deux publics.
Les mesures portant sur la structure de l'ontologie et les mesures basées sur les tâches sont numériques, alors que les métriques portant sur l'utilisabilité de l'ontologie ne le sont souvent pas. Elles peuvent par exemple se présenter sous forme de tableaux explicatifs (Hlomani et al. 2011).
Une dernière approche que nous pouvons signaler pour l'évaluation des ontologies consiste à comparer l'ontologie à évaluer avec des standards reconnus. Ces standards peuvent être des sources de connaissances largement adoptées au sein du domaine décrit dans l'ontologie, ou bien un ensemble de bonnes pratiques permettant de juger de la qualité formelle d'une ontologie. Le manque de couverture ontologique dans le domaine de l'astrophysique rend cette approche difficile à employer.

Nous présentons pour commencer l'évaluation d'ASON au moyen des indicateurs de OntoQA (Tartir et al. 2005), qui est une des seules méthodes dont les critères d'évaluation sont utiles aussi bien à l'utilisateur final qu'au développeur de l'ontologie. Etant donné qu'OntoQA propose de nombreux indicateurs différents, nous choisissons d'exposer ceux que nous jugeons les plus pertinents. Le choix de ces critères est dicté par la volonté de donner un résumé de la structure d'ASON. Ils ont été également choisis pour permettre d'illustrer l'influence des résultats donnés par l'expérimentation au Tableau 11 sur les mesures de qualité. Ces mesures sont essentiellement effectuées pour le module ASTRO-THEM, qui est le module enrichi par la méthode présentée dans ce chapitre.

La population moyenne (Average population, notée P) est définie comme $P = |I|/|C|$, où $|I|$ est le nombre d'individus et $|C|$ le nombre de classes. Le résultat pointe les performances du processus d'extraction de la connaissance. En effet, il indique le nombre moyen d'individus par classe. Si ce nombre est faible, on peut en déduire que certaines classes sont trop spécifiques, ou bien que des individus manquent pour représenter correctement la connaissance du domaine. Si au contraire il est très élevé, il est possible que les classes soient trop générales et manquent de finesse dans la description du domaine.
OntoQA définit l' « Inheritance richness » (IR) ou « richesse d'héritage » comme le nombre moyen de sous-classes par classe. Schématiquement, cet indicateur permet de connaître la verticalité d'une ontologie, un IR fort indiquant la représentation d'une connaissance très détaillé. Une ontologie

horizontale, avec un IR faible indiquera au contraire une ontologie présentant une connaissance générale, variée et qui en conséquence demande à être précisée par l'utilisation de sous-classes.
La Figure 29 illustre l'évolution des valeurs de la qualité moyenne des classes, de la richesse d'héritage et de la population moyenne en fonction de la méthode de mesure de similarité et de la taille de matrices de similarité utilisées dans la méthode d'enrichissement.

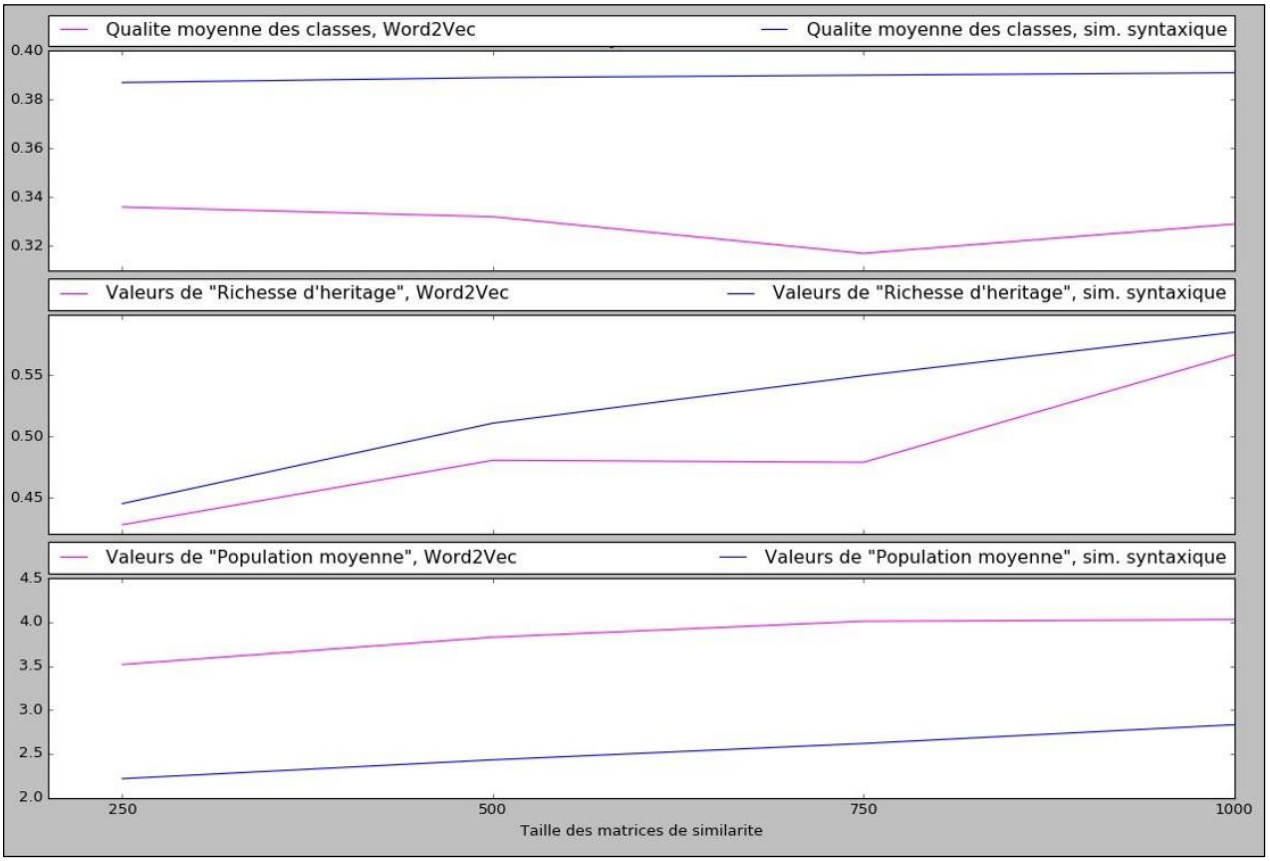


Figure 29: Evolution des métriques en fonction de la taille des matrices de similarité

L'utilisation de la mesure de similarité syntaxique (présentée dans la partie 3.5.3 de ce document pour définir le contenu des matrices de similarité entre descriptions de capacités de services) donne une meilleure qualité moyenne des classes que l'utilisation de Word2Vec. Cela s'ajoute à une richesse d'héritage plus haute et à une population moyenne plus basse. Ces indicateurs montrent que la méthode de construction semi-automatique du module de domaine forme une représentation de la connaissance plus précise en utilisant la mesure similarité syntaxique présentée dans ce document plutôt qu'en utilisant Word2Vec. La qualité moyenne des classes est en effet meilleure, les classes plus spécialisées et les individus par classe moins nombreux.

Un autre indicateur d'importance est le « Class Richness », indiquant comment les instances sont réparties dans les classes. Il est défini comme CR = |C'|/|C|, où |C'| est le nombre de classes possédant au moins une instance et |C| le nombre de classes dans l'ontologie. Dans notre cas CR=1 par définition. Cela est dû au fait que, d'après notre méthode les individus dérivent des clusters dont proviennent

également les classes elles-mêmes, et peuplent les classes identifiées dans leurs clusters d'origine. Ce qui revient à exprimer qu'aucune classe n'est inutile dans le schéma résultant.

Les deux derniers critères issus des propositions de OntoQA qui nous paraissent intéressants sont l' « Attribute Richness » et le « Relationship Richness ».

Nous avons conçu la structure d'ASON de telle façon qu'elle soit réutilisable dans des contextes différents de ceux de l'astrophysique. La population de l'ontologie, issue de l'analyse de descriptions spécifiquement astrophysiques, reste dépendante du domaine d'application. Elle doit cependant rester suffisamment générique pour que sa réutilisation dans des domaines connexes à l'astrophysique, sa lisibilité, son utilité dans d'autres contextes astrophysiques puisse s'envisager. Dans cette optique, l'indicateur de richesse des attributs (Attribute Richness, AR) défini dans OntoQA est un indicateur approprié, qui évalue la qualité de la documentation et de la lisibilité de l'ontologie par un humain.
AR = |att| / |C|
Dans cette définition att représente le nombre d'attributs (qui, dans OntoQA sont les éléments de descriptions en langage naturel associés à un concept) définis pour chaque classe et C représente le nombre de classes.
Pour ASON nous avons AR = 14 lorsque la méthode est utilisée avec une taille de matrices de similarité de 1000 x 1000, formées par mesure syntaxique. C'est une valeur haute, expliquée par le fait que chaque classe, chaque service, chaque entrée et sortie de service décrit dans l'ontologie vient avec une annotation qui est le contenu de la ou des descriptions qui lui sont reliées. De plus, ASON hérite des annotations présentes dans l'ontologie HELIO, des définitions des UCDs et du thesaurus de l'UAT. Il est important que cet indice AR reste élevé tout au long du cycle de vie de l'ontologie, puisqu'il en quantifie le potentiel de réutilisabilité.
La richesse des relations (Relationship Richness, RR) indique dans quelle mesure les relations enrichissent le schéma général de l'ontologie. Le RR est défini comme suit:

$$|R|/(|SC|+|R|)$$

R représente le nombre de relations défini dans le schéma de l'ontologie, SC est le nombre de sous-classes. Etant donné que le module thématique ASTRO-THEM est essentiellement composé de la taxonomie extraite des descriptions de services, cette mesure de RR est plus pertinente quand elle est prise dans le module GEOS qui spécifie l'architecture ontologique des services. En effet, l'enrichissement d'ASTRO-THEM ne crée pas de nouvelles relations dans le schéma, mais se contente d'instancier les relations déjà existantes définies par GEOS. Ainsi spécifiquement pour GEOS, nous avons RR = 91 / (134 + 91) = 40 %.

Peu de méthodes d'évaluation automatiques existent. « Oops ! » (Ontology Pitfall Scanner) (Poveda-villalón et al. 2012), est une de ces méthodes automatiques, qui analyse les défauts des ontologies du point de vue des développeurs. Elle permet de détecter 35 types d'erreurs de conception différents et

propose d'appliquer cette analyse par l'utilisation d'une interface Web[52]. « Oops ! » est un bon complément aux analyses précédentes, qui rendent compte de la structure du schéma de l'ontologie. « Oops ! » s'intéresse aux erreurs techniques, comme par exemple des mauvaises définitions de tansitivité dans les relations ou l'utilisation de conventions de nommage différentes dans une même ontologie.

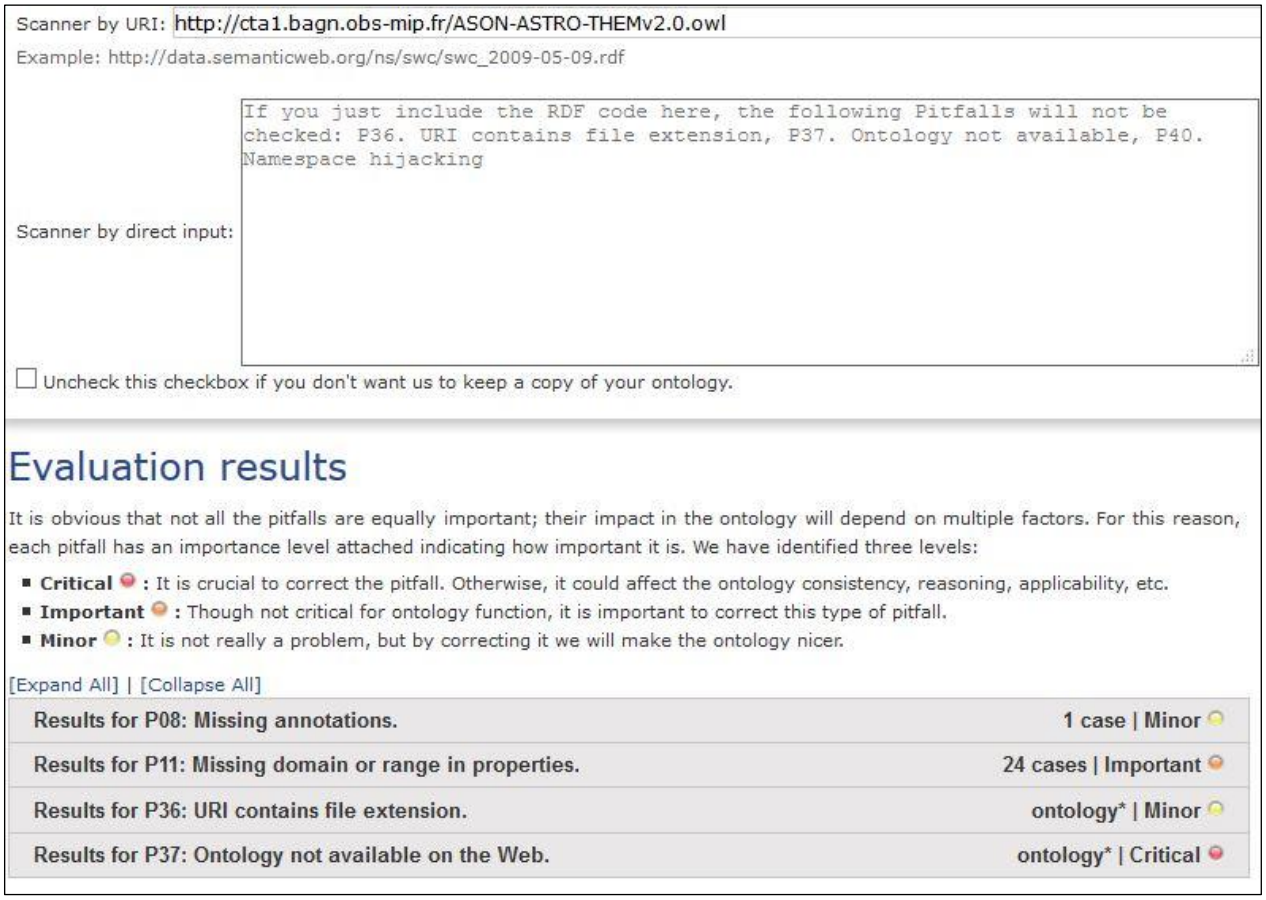


Figure 30: Evaluation de ASTRO-THEM par Oops!

L'interface Web de « Oops ! » pointe 4 erreurs dans l'évaluation d'ASTRO-THEM. Après l'intégration automatique de plus de 11000 services, deux erreurs mineures, une erreur importante reproduite 24 fois et une erreur critique apparaissent. L'erreur critique disparaît quand le module est utilisé à l'intérieur d'ASON, et non plus considéré séparément. Les erreurs qui demeurent n'empêchent pas l'ontologie de remplir son rôle. Elles sont de plus en nombre assez faible comparativement à l'enrichissement effectué pour que l'évaluation de « Oops !» puisse contribuer à valider la méthode d'enrichissement utilisée.

[52] http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/technologies/292-oops/

Pour terminer l'évaluation de l'ontologie, d'autres critères non numériques, présentés dans le Tableau 12 peuvent apporter des précisions utiles (Hlomani & Stacey 2014):

Tableau 12: Critères non numériques d'évaluation pour ASON

Critère	**Evaluation pour ASON**
Adaptabilité	L'adaptabilité dans ASON est assurée par la séparation entre les modules thématiques et le module de description des services GEOS. Réutiliser l'ontologie dans d'autres contextes que le cas d'application présenté dans ce manuscrit revient à repeupler ces deux modules avec les descriptions apportées par le nouveau contexte.
Couverture	Les concepts d'ASON couvrent tous les domaines décrits dans les services de l'IVOA, auxquels s'ajoutent les concepts d'héliophysique décrits dans HELIO.
Couplage	Les concepts de HELIO sont embarqués dans ASON, de sorte que les deux ontologies ne sont pas couplées. Les références externes à OWL-S (Process, Profile et Service) représentent l'héritage de OWL-S présent dans ASON.
Précision	La connaissance décrite dans ASON vient de définitions préexistantes (UCDs, UAT), mais dérive principalement de l'analyse de définitions exprimées en langage naturel. Si des défauts de précision venaient à apparaître, leur origine serait à chercher dans les imprécisions de la méthode d'analyse de ces définitions.

L'évaluation d'ASON passe par l'évaluation des deux modules qui la composent. La méthode générale d'enrichissement proposée forme un schéma cohérent, bien annoté et dont chacune des classes est utilisée.

Le choix de la méthode de mesure de similarité à adopter à l'intérieur de la méthode d'enrichissement générale est toutefois difficile. Conclure sur ce point demandera d'utiliser la méthode dans des contextes variés. Pour notre cas, nous avons utilisé la méthode syntaxique proposée dans ce manuscrit, en raison de la confiance légèrement plus grande accordée aux classes déduites. La population moyenne plus faible est un indicateur de nature variable, et devra être réexaminé au fur et à mesure de l'enregistrement de nouveaux services dans l'ontologie.

3.7. Conclusion

Nous avons présenté dans ce chapitre une méthode pour l'enrichissement d'un module de domaine. Cette méthode permet d'extraire une représentation taxonomique des connaissances à partir de fragments de texte non structurés. Elle minimise la nécessité d'une connaissance ontologique du domaine préexistante, et élimine la nécessité d'une structure grammaticalement cohérente à l'intérieur des textes utilisés comme source principale de connaissance. En utilisant l'apprentissage non supervisé par regroupement des descriptions, nous parvenons à faire émerger une structure de concepts contenue à l'intérieur des capacités de services fournies. Cela est possible par l'utilisation d'un algorithme de *clustering* sur un ensemble de matrices composées des mesures de similarité entre les descriptions. Des techniques de traitement du langage naturel (NLP) appliquées sur les groupes de textes résultant de ce *clustering* permettent d'en dégager des classes et des sous-classes et il est possible de donner une estimation de la confiance accordée à cette extraction.

La méthode proposée ne nécessite que peu ou aucune connaissance à priori des éléments de taxonomie contenus dans cette masse de textes. Toutefois, une connaissance existante même éventuellement peu fournie, est de nature à assurer que l'algorithme identifie correctement un minimum de classes prédéfinies. Bien que la qualité de la structure obtenue ne soit pas parfaite, elle est quantifiée et sa population peut être automatisée avec de bonnes performances.

Les étapes de la méthode exposée dans ce chapitre sont représentées sur la Figure 31.

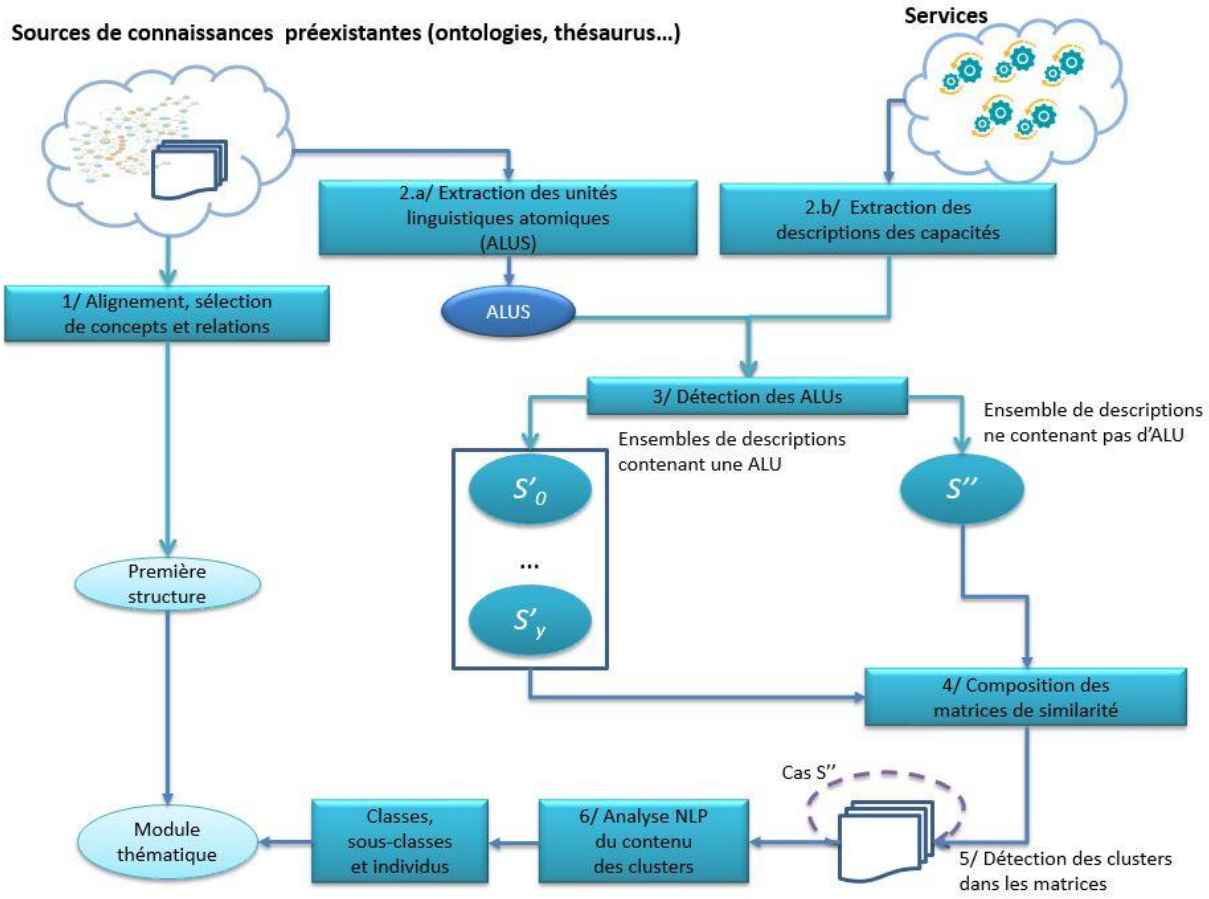


Figure 31: Etapes de la méthode proposée pour la définition semi-automatique d'un module de domaine

Cette méthode n'extrait pas de relations, parce que la tâche d'extraction des relations demanderait des phrases grammaticalement cohérentes que la description des services astrophysique ne fournit pas. En particulier, cette extraction de relations dépend de la présence d'un verbe dans la phrase, dans la plupart des approches.

Le Tableau 13 résume les différences entre les principales approches existantes citées dans l'état de l'art et la méthode proposée ci-dessus pour l'extraction de la structure d'une ontologie à partir de texte. Les critères de comparaison sont la dépendance à la structure grammaticale des textes disponibles, et la dépendance à des sources de connaissance existantes suffisamment couvrantes pour soutenir la méthode d'extraction.

Tableau 13: Comparaison entre la méthode proposée et les approches existantes

Proposition	Dépendance à la structure grammaticale	Dépendance à des sources préexistantes
OLLIE (Mausam et al. 2012)	Critique	Aucune
NELL (Cronin et al. 2011)	Critique	Forte
YAGO (Suchanek et al. 2007)	Faible	Critique
OBIE (Pia 2015)	Forte	Critique
Méthode proposée dans ce manuscrit	Nulle	Faible

Outre les descriptions des capacités des services utilisées pour valider la méthode proposée dans ce manuscrit, l'IVOA fournit pour chaque service une description générique, bien formée et en anglais. Ces descriptions de services sont d'un plus haut niveau d'abstraction que les descriptions de quantités astrophysique précises et ne décrivent pas les capacités des services, c'est pourquoi nous ne les avons pas exploitées dans ce travail. Toutefois, comparer la connaissance extraite de ces descriptions génériques de services avec la structure obtenue par la méthodologie présentée ci-dessus et d'autres ontologies plus globales pourrait aider à l'extraction de relations.

La combinaison du module thématique et du module générique de services forme donc l'ontologie proposée dans ce manuscrit, que nous avons appelée ASON pour le cas astrophysique et dont la structure finale est exposée dans la figure 31.

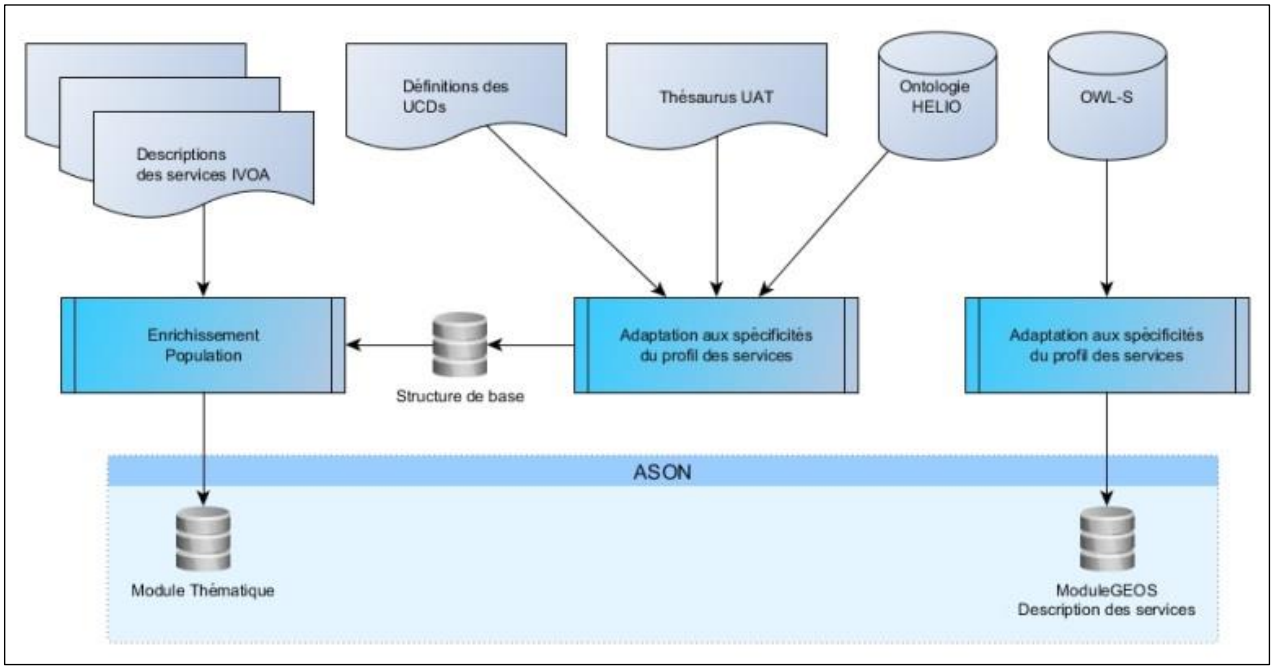


Figure 32: Structure finale d'ASON

Cette ontologie sert de support pour une composition automatique et sémantique des services, qui est détaillée dans le chapitre suivant.

Chapitre 4 : Composition sémantique et automatique de services

4.1. Introduction

La composition des services Web est un sujet de recherche d'actualité en informatique. En quelques années ce domaine a fait de grands progrès ; en passant de peu d'applications concrètes capables de démontrer l'utilité des services Web (Bartalos et al. 2011) à de nombreuses applications pratiques spécifiquement conçues pour des besoins particuliers (Amudhavel et al. 2016). De nombreuses approches abordent la composition de services Web selon des angles différents comme les protocoles de communication, les mécanismes de découverte de services, la qualité de service (QoS), etc (Amudhavel et al. 2016). Il en ressort que la composition est la pierre angulaire des services Web puisqu'elle recouvre de nombreux aspects comme la découverte de services, la sélection et l'exécution. Après l'élaboration du module générique de description de services et la proposition d'une méthodologie pour l'enrichissement et le peuplement d'un module thématique dédié au contexte applicatif, ce chapitre a pour vocation de présenter notre processus de composition automatique de services sémantiques. Pour cela, dans un premier temps, nous exposerons les exigences fonctionnelles et non fonctionnelles imposées par le profil des services du contexte applicatif, et les approches de composition existantes. Par la suite, nous identifierons les limites de ces approches vis-à-vis de ces exigences. Enfin, nous détaillerons le processus proposé, son fonctionnement général et les algorithmes sur lesquels il repose.

4.2. Exigences imposées par les services

De nombreux services Web (plus de 10 000) sont disponibles dans l'OV. Malheureusement, l'adaptation des approches existantes pour la composition des services à l'architecture de l'OV est difficile. Tout d'abord, l'OV ne repose pas sur des technologies habituelles en informatique telles que WSDL, SOAP et UDDI mais utilise ses propres formats et protocoles; ce qui rend les méthodes de composition de SWS existantes difficiles à adapter.

Une autre difficulté vient de la façon dont les services sont utilisés, c'est-à-dire de leurs exigences fonctionnelles. La plupart des services astrophysiques sont atomiques, sans états avec une combinaison inhabituelle de leurs paramètres d'entrée. Un service peut être appelé avec un ensemble de valeurs d'entrée obligatoires qui peuvent ou non être complétées par des entrées supplémentaires. Pendant la composition, un service est sélectionné en fonction de l'utilité de ses sorties et de la disponibilité de ses valeurs d'entrée. Par conséquent, le fait qu'une entrée spécifique soit obligatoire ou non pour un service donné doit être connu au cours de la composition.

Les entrées des services peuvent parfois être fortement connectées, de sorte qu'un ensemble d'entrées d'un service doive provenir d'une source unique. Un exemple de cette particularité est rencontré lorsqu'un service a besoin d'une valeur de mesure et de la barre d'erreur liée à ladite mesure. La mesure elle-même et sa barre d'erreur perdent toute leur signification si elles ne proviennent pas de la même source.

Les aspects observationnels déjà évoqués de l'astrophysique signifient qu'un service ne fournit des résultats que pour les observations effectivement réalisées avec l'instrument auquel ce service est lié. Lorsqu'un service est décrit comme fournissant une sortie (par exemple, la température d'une étoile)

pour une entrée (par exemple, le nom de l'étoile), il n'est pas certain que le service fournira effectivement la sortie pour chaque étoile. Le service ne fournira cette information que pour un ensemble d'étoiles, celles qui ont été observées par l'instrument auquel le service est lié. Pendant la composition, cela implique que l'on ne peut jamais être sûr que les sorties d'un service seront garanties. Cela dépend du contexte global de la composition, qui doit donc être représenté pour assurer la meilleure garantie de succès. Ce contexte peut être exprimé par l'historique de la composition et les commentaires des utilisateurs.
La proposition d'une composition automatique du service Web sémantiques pour l'astrophysique nécessite que les exigences fonctionnelles et non fonctionnelles mentionnées ci-dessus soient prises en compte. La composition doit assurer que les entrées obligatoires pour les services sont fournies, ainsi que toute autre entrée disponible. La sélection des services doit également tenir compte des éléments de contexte tels que l'historique de la composition et les commentaires des utilisateurs afin de maximiser la probabilité de succès de la composition. Cette composition ne doit pas dépendre de technologies habituelles que l'OV n'utilise pas. Le présent chapitre propose un processus de composition des services qui répond à ces objectifs.

4.3. Etat de l'art

Dans une étude récente de la littérature au sujet de la composition de services Web (Lemos et al. 2016), un cadre formel pour l'analyse des approches existantes a été proposé. Douze plateformes de composition trouvées dans des revues, des conférences et les propres connaissances des auteurs de l'étude ont été analysées. Ces plates-formes ont été choisies en fonction de leur «pertinence, importance, impact et originalité de l'approche». Parmi toutes les considérations exposées dans cette enquête, nous pouvons souligner trois points importants:

- Les approches basées sur SOAP et REST restent les plus nombreuses, avec 8 sur 12 plates-formes utilisant ces protocoles. Une seule approche basée sur une ontologie de services a été étudiée : Simple Hierarchical Ordered Planner (SHOP)2 (Batista 2011). Cela montre que les compositions SWS ciblant des cas réels et des résultats validés sont peu nombreuses et restent donc un créneau pour la recherche.
- Seules trois approches de composition scientifique ont été mentionnées dans cette enquête et seule la plate-forme Taverna a fait l'objet d'une enquête approfondie avec le cadre proposé.
- Aucune des approches n'a été classée comme ciblant «l'utilisateur final» qui, dans le vocabulaire de l'enquête, désigne un utilisateur novice sans connaissances techniques. Chaque plateforme nécessitait un certain niveau de compétence en programmation. La plate-forme Taverna permet à un utilisateur sans expérience technique de télécharger et d'exécuter un workflow. Cependant, la modification du workflow reste hors de portée pour un tel utilisateur. Ce manque de méthodes de composition ergonomiques a déjà été souligné dans l'enquête de Bartalos et Bielikova (Bartalos et al. 2011).

Les Services Web Sémantiques (McIlraith et al. 2001) utilisent des langages spécialement destinés à une expression sémantique, comme DARPA Agent Markup Language (DAML-S), dont OWL-S est une évolution, complétés par des langages de programmation logique pour l'exécution du raisonnement. La composition des services Web sémantiques est un sous-domaine de la composition des services Web. Bien que cette dernière connaisse des succès concrets, la première reste principalement théorique, car les SWS n'ont pas encore été largement adoptés malgré l'existence d'ontologies dédiées comme OWL-S et WSMO (Pedrinaci & Domingue 2010). Une des raisons expliquant la lente adoption des SWS est la difficulté rencontrée pour annoter les services existants pour les convertir en SWS (Tosi & Morasca 2015), (Pedrinaci & Domingue 2010). Cet état de fait est illustré par l'enquête dans (Lemos et al. 2016) qui ne comprend qu'une approche SWS sur douze approches analysées. Néanmoins, les recherches sur l'utilisation de services Web sémantiques proposent des solutions pour la composition de SWS. Ces recherches peuvent porter soit sur des aspects spécifiques de l'ensemble du processus de composition, soit sur l'ensemble du processus qui comprend généralement une étape de sélection de services basée sur des paramètres fonctionnels.
La sélection d'une composition parmi toutes les compositions disponibles, basée sur des paramètres non fonctionnels, est la « Quality of Service », QoS. Dans les travaux de Zhao et al (Zhao et al. 2013) les paramètres non fonctionnels sont hiérarchisés en utilisant la logique floue et les compositions possibles sont triées en fonction de cette hiérarchie. La hiérarchisation de chaque service dans la composition à l'égard de chaque paramètre non fonctionnel mène au classement des compositions. Ce classement est obtenu grâce à des calculs de Pareto-dominance et de distance de Tchebycheff entre les solutions candidates.

L'approche iServe (Pedrinaci et al. 2010) aborde la découverte de services Web par le partage des annotations sémantiques associées à chaque service. Ces annotations sont accessibles par différents moyens : Une API RESTFul dédiée, des requête sur des triplets RDF ou un raisonnement SPARQL. iServe assure donc le rôle de UDDI dans l'approche UDDI/WSDL à travers l'utilisation d'annotations sémantiques.

Une autre utilisation de SPARQL pour la découverte de services Web est proposée dans (Sbodio et al. 2010), qui met l'accent sur l'expression des pré-conditions et des post-conditions à l'invocation des services décrites dans OWL-S. Cette approche suppose qu'un registre de services soit disponible (registre qui peut être un distant). Chaque ontologie OWL-S doit également partager la même référence de termes pour exprimer les descriptions et les objectifs des services. Ce travail ne traite pas des «conditions minimales suffisantes», c'est-à-dire des conditions préalables indispensables pour l'exécution d'un service, et de celles qui améliorent cette exécution. Les aspects non fonctionnels des services (qualité de service) ne sont pas non plus abordés.

Dans (Puttonen et al. 2013), Puttonen et al. utilisent également SPARQL et OWL-S pour composer des SWS. Cette approche présente non seulement le processus de composition, mais aussi l'invocation des services dans la composition. La composition vise alors à atteindre un objectif qui est un état du système remplissant les exigences de la composition. L'algorithme de composition utilisé fonctionne en chaînage direct, allant des préconditions disponibles au début de la composition (l'état initial) et explorant chaque chemin disponible de cet état initial vers l'état d'objectif. Un argument optionnel

permet à l'algorithme de se concentrer uniquement sur les espaces d'état qui font progresser la composition vers l'état d'objectif, de telle sorte que tous les espaces d'états ne soient pas explorés. Cette approche ne traite pas de la réutilisation des compositions précédentes, ni par morceaux ni comme un tout. Les aspects QoS ne sont pas non plus traités.

L'utilisation d'un historique des compositions passées est explorée dans (Xing & Wenpeng 2010), proposant un algorithme de composition par chaînage arrière (appelé «de droite à gauche» dans leur papier) pour des services à sortie unique et entrées multiples. La principale difficulté rencontrée est la difficulté de fournir une ontologie de domaine efficace pour la description des services Web. Le QoS n'est pas non plus abordé.

Certaines méthodes de composition prennent en compte l'aspect QoS. Rodriguez-Mier et al. (Rodriguez-Mier et al. 2016) proposent un algorithme de composition en chaînage avant, basé sur iServe pour la découverte de services. Dans cette approche, chaque combinaison de services à même d'atteindre l'objectif est générée lors d'une première étape. Ensuite, un deuxième algorithme est exécuté sur ces combinaisons pour trouver la combinaison de coût minimum. Les critères d'établissement du coût peuvent varier (nombre de services, paramètres de QoS ...).

Bansal et ses co-auteurs (Bansal et al. 2014) proposent une approche du problème de composition des SWS par le développement d'un moteur de découverte et de composition de service écrit en Prolog, la base de faits découlant de la description sémantique des services exprimés en USDL (Simon et al. 2005). La découverte des services résulte d'une mise en correspondance entre les descriptions de services USDL, la description des exigences de composition exprimée également en USDL et WordNet (Miller 1995), ou une autre ontologie spécifique au domaine. WordNet ou l'ontologie de domaine sert de référence pour la sémantique des concepts. Le choix des services repose sur la mesure de leur «degré de centralité» représentant le nombre de services tiers avec lesquels un service donné est en contact. Cela implique qu'une description sémantique exprimant la description des liens existants entre les services soit disponible, les services obtenant la centralité la plus élevée étant considérés comme de meilleurs choix au cours de la composition. Les compositions résultantes sont décrites en OWL-S. Cette approche n'aborde pas la phase d'exécution de la composition; son résultat est la composition elle-même et non le résultat de l'exécution.

Le processus présenté dans de chapitre englobe toutes les étapes de la composition et intègre les retours des utilisateurs et l'historique des compositions dans l'évaluation de la QoS. Aucun langage ni format spécifique n'est imposé pour l'expression des exigences, et la découverte et la composition des services utilisent une seule ontologie pour la description des services. Ce processus est validé dans le contexte de l'astrophysique, et ses principes restent valables pour d'autres applications.

4.4. Motivations

L'enquête menée par Lemos et al. (Lemos et al. 2016) n'a retenu que SHOP2 en tant qu'approche compatibles SWS, et uniquement Taverna comme approche abordant les workflows scientifiques. D'après notre lecture des recherches entreprises dans le domaine, il n'existe actuellement aucune

approche proposant des solutions adaptées à la composition des services Web sémantiques en astrophysique. Pourtant, l'astrophysique est un bon exemple d'un domaine scientifique où les SWS peuvent faciliter la composition des services.

L'astrophysique est un cas de composition sémantique de services dans lequel une DCI spécifique propose une approche intégrée, permettant de décrire les formats et les modèles de données utilisés par les services Web. Plusieurs services sont également disponibles en-dehors de cette DCI, qu'il s'agisse de services permettant de récupérer des données, mais aussi des services d'analyse, en ligne ou installés sur des machines locales. En conséquence, de nombreux services Web (plus de 10 000 seulement pour le protocole OV le plus utilisé) sont disponibles pour l'astrophysique, à l'intérieur comme à l'extérieur de l'architecture OV. Cette quantité de services rend leur composition à la fois nécessaire et complexe. Elle est nécessaire, parce que l'utilisation conjointe de certains services au sein de ces multiples possibilités peut produire des résultats combinant les capacités de plusieurs services individuels. Elle peut également se servir de résultats fournis dans des services comme paramètres d'entrée pour d'autres services et ainsi espérer enchaîner les traitements et les requêtes pour produire des résultats utilisant au mieux le maximum de compétences disponibles.
Elle est complexe, car la multiplicité des usages contenus dans les descriptions, des unités disponibles et des spécificités d'observation de chaque service n'est que partiellement couverte par les mécanismes actuels disponibles dans l'observatoire virtuel.

ASON nous donne l'opportunité d'accéder aux services Web astrophysiques comme aux services d'analyse en utilisant une description ontologique basée sur OWL-S. Cela ramène le problème de composition des services à un problème de composition de services sémantiques. La mise en place d'une interface utilisateur conviviale est obligatoire pour que la composition de SWS en astrophysique atteigne son maximum de potentiel. Cela signifie qu'il faut chercher à atteindre les objectifs suivants :

- Fournir à l'utilisateur un moyen simple pour exprimer les exigences de la composition
- S'assurer que la composition a les meilleures chances de réellement retourner un résultat pour une demande spécifique
- Exécuter les compositions résultantes, afficher les résultats et permettre une recomposition rapide des services en modifiant leurs critères de sélection.

Exprimer les exigences de la composition d'une manière simple pour l'utilisateur demande de réunifier le langage naturel avec la description sémantique des sorties et des entrées des services. Il est nécessaire d'avoir un système plus flexible qu'une simple correspondance de mots clés qui ne reconnaîtrait pas les mots-clés non enregistrés dans un référentiel donné. Un système à base de mots-clés courrait également le risque de transformer les habitudes d'experts exprimées dans le référentiel de services en normes pour de futures descriptions (par exemple, exprimer «k-band magnitude» pour «magnitude in band k» ou vice versa). L'expression naturelle des exigences de la composition devrait également être plus expressive que la définition d'un AtomicConcept trouvée dans USDL. La méthode d'enrichissement et de population présentée au Chapitre 3: Construction semi-automatique d'un module de domaine permet de peupler l'ontologie d'individus annotés, dont les annotations sont une description en langage

naturel de la quantité que l'individu représente. Nous cherchons à ce que l'expression naturelle des exigences de la composition soit mise en correspondance avec les expressions naturelles contenues dans ASTRO-THEM.

Assurer la qualité de la composition est un enjeu critique. Etant donné que chaque service a sa propre spécificité, il y a deux paramètres importants à prendre en compte:

- La spécialisation du service (quelles informations, pour quelle qualité individuelle le service est-il susceptible de fournir ?).
- La disponibilité réelle de l'information, dans un service donné et pour une entrée donnée.

La spécialisation du service ne peut être confirmée que par l'utilisateur, de sorte qu'il est nécessaire de fournir à l'utilisateur la possibilité d'évaluer chaque service pour les données renvoyées. Il est également nécessaire de tenir compte de ce retour des utilisateurs pour les compositions futures dès qu'ils sont exprimés, sans attendre une mise à jour de l'ontologie ou un autre mécanisme retardant cette prise en compte.

La phase d'exécution, qui est la dernière phase de la composition de services concerne l'orchestration des services et l'affichage des résultats. Une extraction d'informations (qualité de service, évaluation du contexte et des collaborations, …), concernant le déroulement de cette phase, est effectuée. En effet, pour optimiser le déroulement de la phase d'exécution, il est judicieux d'enregistrer pour des compositions futures le fait qu'un service fournisse des informations pertinentes dans un contexte donné. Cela introduit un niveau de préférence pour ce couple service/contexte, vis-à-vis d'autres services pour lesquels le contenu serait inconnu dans ce contexte. Il est également judicieux de conserver, indépendamment du contexte, la liste des services ayant collaboré à la production d'informations pertinentes. Cela permet d'introduire un second niveau de préférence privilégiant l'utilisation de services ayant déjà été utilisés ensemble avec succès. L'indication faite par l'utilisateur de la qualité des données fournies par un service est également une information importante. La sauvegarde de ces informations crée des variables d'action sur le processus, qui modifieront le comportement de ce dernier dans des proportions paramétrables. L'influence de la variable mesurant l'adéquation service/contexte par rapport à la celles mesurant la qualité estimée ou les précédentes utilisations conjointes des services pourra ainsi être fixée par l'utilisateur.

4.5. Processus de composition proposé

Une vue générale du processus de composition proposé dans ce chapitre est présentée en Figure 33.

Les trois phases composant l'approche présentée dans ce chapitre sont l'identification des exigences, la phase de composition et la phase d'exécution des compositions produites. La phase d'évaluation de la qualité de service est incluse dans la phase de composition, mais mérite d'être détaillée individuellement.

L'identification des entrées et des sorties de la composition consiste à rapprocher les informations demandées par l'utilisateur des informations décrites dans l'ontologie (dans le module thématique de l'ontologie). Cette identification permettra, lors de la phase de composition, de sélectionner comme

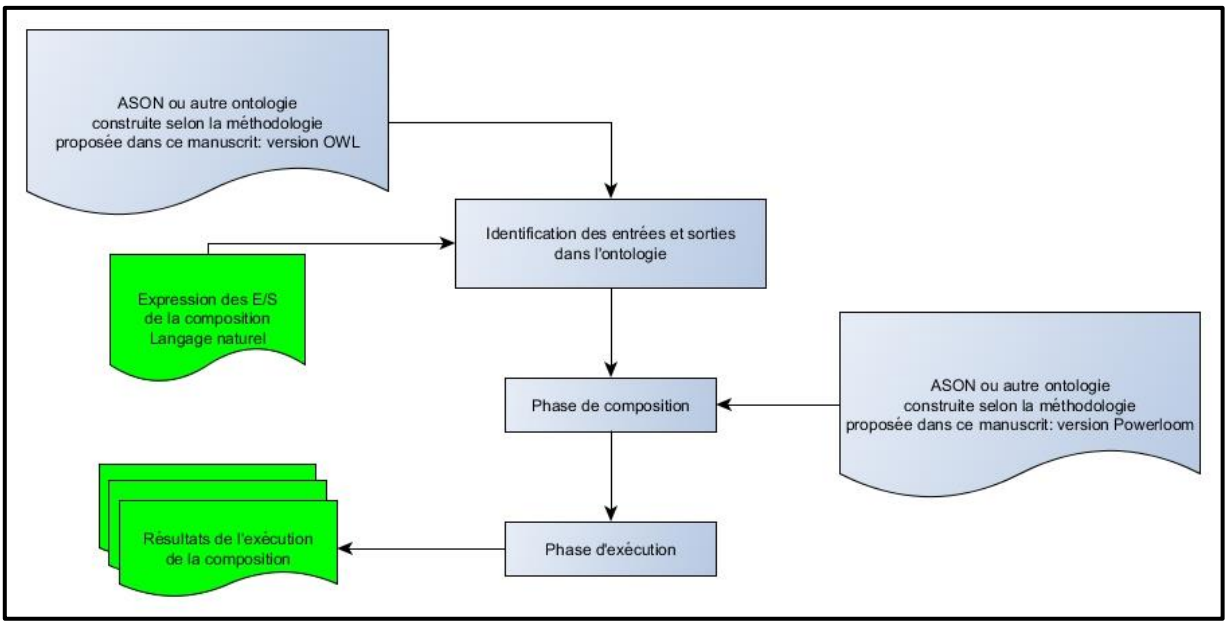


Figure 33: Vue générale du processus de composition proposé

point de départ uniquement les services à même d'apporter des éléments de réponse au problème posé. La phase de sélection de services n'est pas dissociée dans notre proposition de la phase de la composition proprement dite. L'identification des exigences porte naturellement sur l'identification des informations demandées par l'utilisateur, mais elle porte également sur l'identification des informations données par le même utilisateur. En effet, l'utilisateur donne un ou plusieurs points de départ (par exemple, le nom d'un objet) qui sont les seules informations disponibles pour initier la composition. Ces points de départ doivent être identifiés dans l'ontologie. Schématiquement, nous pouvons dire que nous envisageons les exigences de l'utilisateur comme un méta-service possédant ses entrées (les informations fournies) et ses sorties (les informations demandées). La tâche de la composition revient à construire ce méta-service à partir des services individuels décrits dans l'ontologie.

La phase de composition recherche, à partir des individus de l'ontologie correspondant aux informations de sortie demandées, quels sont les services fournissant ces individus en sortie. Puis, de proche en proche elle sélectionne les services fournissant en sortie les entrées nécessaires à ces services identifiés, et ainsi de suite. Le résultat de cette phase est donc une liste de chaînes de services possibles, qui sont tous les workflows disponibles. L'évaluation de la qualité des services se fait durant cette phase.

La dernière phase est la phase d'exécution, qui planifie l'appel aux services en fonction des workflows identifiés et de la qualité estimée de chaque service. Après avoir planifié ces exécutions, elle gère les appels aux services et la production des résultats de ces appels. L'utilisateur a alors la possibilité de

changer le poids de l'influence affectée à chacun des paramètres de qualité pour revenir sur la composition des workflows et essayer des combinaisons de services différentes. Cette phase d'exécution est la phase qui présente le plus d'interaction avec l'utilisateur.

La Figure 33 présente une vue générale du processus que nous proposons. Les éléments représentés en vert sur cette figure représentent les éléments accessibles à l'utilisateur.

La Figure 34 fait apparaître les éléments internes des trois phases principales de la composition que nous proposons. Ce chapitre décrira le rôle et le fonctionnement de ces trois phases.

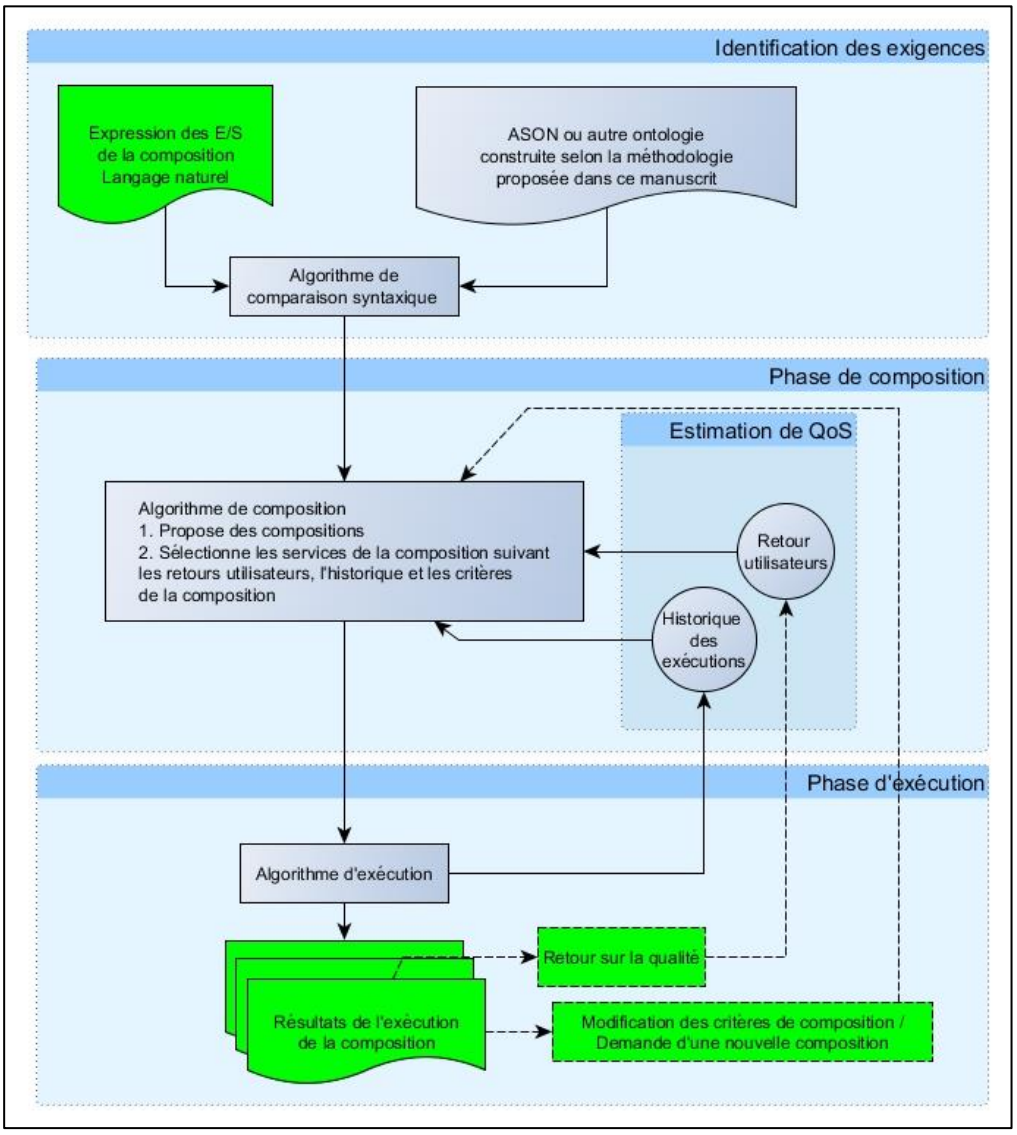


Figure 34: Détail du processus de composition proposée

La sous-phase d'évaluation de la qualité de service (QoS), contenue dans la phase de composition proprement dite sera également décrite dans ce chapitre.

4.5.1. Phase d'identifications des exigences

ASON contient les descriptions des services comme la description du domaine d'application, et joue par conséquent le double rôle d'ontologie de description des services et de registre de services. La phase de sélection des services utilise les exigences de l'utilisateur concernant la composition comme paramètres d'entrée. Ces exigences comprennent:

- Les sorties recherchées qui doivent être le résultat de la composition, exprimée en langage naturel
- Les informations d'entrée que l'utilisateur peut fournir, exprimées en langage naturel
- Les valeurs associées aux informations d'entrée

La première étape de la phase de sélection est d'identifier les individus dans ASON qui correspondent le mieux à l'expression en langage naturel des exigences de composition. Les individus d'ASON sont associés à des annotations qui sont la description en langage naturel de leur sémantique. Un individu peut avoir plus d'une annotation, chaque annotation correspondant alors à une description différente pour exprimer la même réalité astrophysique. En conséquence, chaque entrée et chaque sortie de la composition doit être liée à l'annotation la plus pertinente d'un individu identifié dans l'ontologie. Les services capables d'utiliser les entrées données et capables de produire les sorties requises peuvent alors être identifiés.

L'identification des individus pertinents repose sur une correspondance syntaxique entre les exigences d'entrées / sorties fournies par l'utilisateur et les annotations des individus ASON. Une valeur de similarité est mesurée, qui exprime la similitude entre chaque description d'entrée / sortie dans la composition (description d'information en langage naturel) et chaque annotation de l'individu en ASON (description d'information en langage naturel). Pour obtenir une telle valeur de similarité, nous utilisons la mesure de similarité syntaxique décrite au chapitre 3.

4.5.2. Phase de composition de workflows

Dans notre approche, la sélection et la composition ont lieu dans une même phase. Les données de sortie de composition et les données disponibles avant la composition sont exprimées par les individus aux annotations les plus approchantes ; toutes les compositions possibles sont alors recherchées en se basant sur les données disponibles et les données de sortie de composition requises.
Cette composition utilise les mécanismes spécifiques à ASON discutés dans le « Chapitre 2: Un module ontologique générique pour les services ». Le premier de ces mécanismes vise à exprimer les entrées et les sorties des services non seulement par leur sémantique, mais aussi par leur mode de représentation. Ce mode de représentation comprend une unité (Jy, degré ...), et un format (ASCII, Flexible Image Transport System (FITS), VOTable ...). Ceci est obligatoire car certains services ne peuvent accepter qu'une combinaison spécifique d'une sémantique, d'une unité et d'un format comme entrée ou fournissent une combinaison spécifique comme sortie. Par conséquent, les entrée et sorties

des services dans notre algorithme de composition de workflow ne sont pas des paramètres tels que «BookTitle» ou «HotelConformationNum» mais des triplets (information, unité, format).
Le deuxième mécanisme relie deux ou plusieurs informations ensemble dans le sens où elles doivent être fournies par le même service pour garder une cohérence. Si un service définit comme entrées une mesure et la barre d'erreur sur cette mesure, cette mesure et sa barre d'erreur doivent évidemment provenir de la même source. Plus généralement ce mécanisme permet, chaque fois que nécessaire, de maintenir la cohérence entre les éléments dont la sémantique peut être très différente mais dont les conditions générales d'observation doivent rester les mêmes.
La troisième considération importante est que certains services (en particulier les services analytiques) peuvent être exécutés avec un ensemble minimum donné d'entrées, mais fournissent parfois des résultats meilleurs ou plus précis avec un ensemble de données d'entrée plus important. C'est pourquoi il est nécessaire de séparer les entrées obligatoires (qui sont nécessaires pour le service à exécuter) des entrées non obligatoires (qui peuvent affiner les résultats du service).
L'algorithme utilisé pour la composition des services basée sur ASON a également ses propres caractéristiques, indépendamment de la spécificité de domaine. Tout d'abord, nous nous plaçons dans le cas où une DCI préexistante fournit au domaine d'application des protocoles d'accès aux données. Les protocoles les plus utilisés dans notre application astrophysique sont le protocole simple d'accès aux spectres (Simple Spectrum Access Protocol, SSAP) pour les spectres, le protocole simple d'accès aux images (Simple Image Access Protocol, SIAP) pour les images, ConeSearch pour une recherche autour des coordonnées dans le ciel, etc. Ces protocoles partagent le même ensemble d'entrées: des coordonnées sur le ciel exprimées en degrés décimaux, un rayon autour de ces coordonnées également exprimé en degrés décimaux. ASON ne se limite pas aux protocoles OV et décrit les services non conformes à l'OV et les services analytiques. Néanmoins, en utilisant un algorithme de chaînage avant sélectionnant parmi les entrées disponibles, tous les services pouvant être utilisés à partir de ces entrées reviendrait à sélectionner tous les services conformes à la norme OV dès que les coordonnées et le rayon sont disponibles dans la base de connaissances. C'est pourquoi nous utiliserons un algorithme de chaînage arrière ; qui ne sélectionnera que des services fournissant des informations utiles dans la composition en ajoutant à la liste des informations demandées toutes les entrées nécessaires pour chaque service utile rencontré jusqu'à ce que les chaînes de composition complète soient composées. Les services avec une chaîne incomplète (avec des entrées manquantes) seront exclus de la composition, tandis que les services avec des entrées complètes formeront les chaînes finales de composition.
Les entrées de l'Algorithme 2 (algorithme d'amorçage) sont les informations requises en sortie de composition et les informations fournies par l'utilisateur (les individus de l'ontologie résultant de la phase d'identification des besoins), ainsi que leurs unités et formats respectifs. Si aucune unité spécifique et / ou aucun format spécifique n'est donné, chaque unité / format disponible est considéré comme satisfaisant. La sortie de cet algorithme est une liste de services qui fournissent les sorties requises pour la composition (individus identifiés avec des unités et formats pertinents, ligne 7 de l'algorithme), accompagnés d'une mise à jour de la base de connaissances (lignes 10 et 11).

L'Algorithme 3 (algorithme de composition) est exécuté sur la base de connaissances mise à jour par l'algorithme d'amorçage en partant de la liste des services générée par l'algorithme de base. La sortie de l'algorithme de composition est un fichier contenant la liste des chaînes de services qui répondent aux exigences de la composition (mise à jour aux lignes 18, 19 et 24 de l'algorithme) avec leurs paramètres associés (mis à jour à la ligne 27). Cette liste intègre toutes les compositions disponibles pour ces exigences. Il s'agit d'un graphe acyclique dirigé dont les arcs sont non pondérés. Dans certains cas, il peut arriver que les services enregistrés dans le fichier de composition après l'exécution de l'algorithme de composition soient inutiles pour la composition réelle. Cela se produit dans un cas très spécifique, quand un service a plusieurs prédécesseurs et que:

- Certains de ces prédécesseurs peuvent être utilisés dans la composition, tandis que d'autres ne le peuvent pas (par manque d'informations d'entrée)
- Les sorties fournies par l'ensemble des prédécesseurs utilisables ne répondent pas à toutes les exigences d'entrées pour le service

Dans ce cas particulier, le service est correctement exclu de la composition, mais les prédécesseurs qui peuvent être utilisés sont élus en tant que services participant à la composition. S'ils ne fournissent aucune information utilisable par un autre service dans la composition, ils deviennent inutiles. Ceci est détecté et résolu par l'algorithme EliminateUselessServices, utilisé dans la phase d'évaluation de QoS.

La base de connaissances que nous utilisons pour la composition des services est une version d'ASON traduite en langage *Powerloom*. Cette base de connaissances est exempte de toute annotation et de toute relation de hiérarchie entre les concepts, puisque les individus concernés pour la composition ont été identifiés dans la phase d'identification des besoins. Chaque composition commence par charger la base de connaissances pour sa propre exécution. Les mises à jour effectuées pendant la composition ne sont pas enregistrées. Le mécanisme de *grounding* d'ASON étend les possibilités de *grounding* d'OWL-S en apportant la notion de «protocole». Un protocole exprime un ensemble d'informations d'entrées obligatoires pour utiliser chaque service en utilisant un protocole commun et une mise en forme commune pour des requêtes à tous les services utilisant ledit protocole. L'information concernant l'URL d'accès à un service est également contenue dans ASON.

Certains services, en particulier les services d'analyse, ne sont pas basés sur le Web et nécessitent une installation sur place, sur le serveur local hébergeant le système de composition. Lorsqu'un service a besoin d'un logiciel spécifique pour être correctement interrogé et ses résultats correctement exprimés; le logiciel en question est également enregistré dans ASON et placé sur un espace disque du serveur où il peut être utilisé. Ces morceaux de code sont appelés «QuerySoftware» dans le module ASON OWL-S, GEOS (GEneric Ontology for Services). Le concept QuerySoftware désigne le logiciel qui interroge un service donné et analyse ses résultats. Dans notre cas, chaque protocole OV est associé à un individu QuerySoftware. Ce logiciel forme la requête avec les entrées et l'URL d'un service, transmet la requête au service, analyse ses résultats et les rend disponibles pour la composition. Pour les services analytiques tels que les bibliothèques scientifiques, ce QuerySoftware est la bibliothèque elle-même.

Algorithme 2: Algorithme d'amorçage

	Entrées:
1:	*W = {(I_0,U_0,F_0)…(I_x,U_x,F_x)} Individu, Uniét, Format: informations demandées comme résultat de la composition*
2:	*G = {(I_0,U_0,F_0,V_0)…(I_x,U_x,F_x,V_x)} Individu, Unité, Format: informations fournies par l'utilisateur*
	Sorties:
3:	*P = {s} liste de services*
4:	*Mise à jour de la base de connaissances (Knowledge Base, KB)*
5:	S = {}
6:	For each (I,U,F) in W:
7:	S <= S + getServicesProviding(I,U,F)
8:	For each (I,U,F,V) in G:
9:	Available(I)
10:	HasDisposableUnit(I,U)
11:	HasDisposableFormat(I,F)

Algorithme 3: Algorithme de composition de services

	Entrées:
1:	*S = {s} liste de services*
	Sorties:
2:	*Fichier F contenant une liste ordonnée de services satisfaisant aux exigences de la composition, avec leurs entrée et sorties respectives*
3	*Fichier F_p contenant les paramètres des services (URL dans les cas les plus fréquents, et l'élément QuerySoftware relié au service)*
4:	**Représentation interne:**
5:	*MP = {(I_0,U_0,F_0)…(I_x,U_x,F_x)} Individu, Unité, Format des entrées obligatoires pour un*
6:	*service*
7:	*OS = {(I_0,U_0,F_0)…(I_x,U_x,F_x)} Individu, Unité, Format des sorties d'un service*
8:	*IS = {(I_0,U_0,F_0)…(I_x,U_x,F_x)} Individu, Unité, Format des entrées d'un service (obligatoires,*
9:	*ou non)*
10:	For each s in S:
11:	P = SeekPredecessors(s)
12:	If P = ! {0} :
13:	S <= S + p
14:	If P = = {0} :
15:	MP = GetMandatoryParams(s)
16:	If (Available(I) & HasDisposableUnit(I,U) & HasDisposableFormat(I,F)) for each
17:	(I,U,F) in MP:
18:	OS = GetOutputs(s)
19:	IS = GetIntputs(s)
20:	For each (I,U,F) ∈ IS:
21:	Write(F,(I,U,F))
22:	Write(F,s)
23:	For each (I,U,F) ∈ OS:
24:	Available(I)
25:	HasDisposableUnit(I,U)
26:	HasDisposableFormat(I,F)
27:	Write(F,(I,U,F))
28:	QS = GetQuerySoftware(s)

29:	URL = GetUrl(s)
30:	Write(F_p,QS,URL)

Algorithme 4: Recherche des services prédécesseurs

	Entrée:
1:	*Service s*
	Sorties:
2:	*P = {p} liste des prédecesseurs pour s*
	Représentation interne:
3:	*CI = {(I_0,U_0,F_0)…(I_x,U_x,F_x)} Individu, Unité, Format pour les entrées corrélées d'un service*
4:	*NCI = {(I_0,U_0,F_0)…(I_x,U_x,F_x)} Individu, Unité, Format Pour les entrées non-corrélées d'un service*
5:	CI = GetCorrelatedInputs(s)
6:	NCI = GetNonCorrelatedInputs(s)
7:	If CI != {} :
8:	Pred = getServicesProviding(CI_n, ∀ n = 0.. |CI|)
9:	P <= Pred
10:	If NCI != {} :
11:	Pred = getServicesProviding(NCI_n, n = 0.. |NCI|)
12:	P <= P + Pred

4.5.3. Evaluation de la qualité de service (QoS)

Le contenu des services astrophysiques dépend beaucoup du contexte dans lequel ils sont utilisés. La plupart du temps, un service sémantiquement décrit comme fournissant une certaine sortie ne fournira pas réellement ladite sortie pour un ensemble donné de valeurs d'entrée. La raison peut être que l'analyse ne peut pas être effectuée par le service pour ces valeurs (service d'analyse, bibliothèque locale de traitement), soit parce que la cible demandée n'a pas été observée (service fournisseur de données). En conséquence, chaque service candidat pour la composition doit être retenu pour participer à des compositions alternatives, et non retiré du système même si un autre service apporte une meilleure qualité globale de la composition. Lorsqu'un workflow ne fournit pas une sortie demandée, l'utilisateur doit pouvoir modifier les critères de sélection de la composition et exécuter une autre combinaison de services. L'utilisateur doit également pouvoir exécuter toutes les compositions possibles et voir les résultats. Que l'on choisisse un ensemble différent de critères et qu'une autre composition soit exécutée, ou que chaque composition soit exécutée, nous évitons en retenant chaque service candidat la répétition des phases de sélection et de composition qui précèdent la sélection d'un service par rapport à un autre. Notre approche automatise la sélection non fonctionnelle de services (QoS) pour la meilleure composition possible. Les critères de sélection des meilleurs services sont basés sur les retours d'expériences des utilisateurs et l'historique d'exécution. L'historique d'exécution conserve la trace de deux paramètres:

- Les valeurs d'entrée des compositions pour lesquelles un service a produit un résultat utilisable

- Une liste de tous les services candidats pour le workflow courant, précédemment employés conjointement dans des workflows qui ont produit un résultat satisfaisant

Ces deux paramètres (tous les deux sauvegardés dans un fichier à la ligne 29 de l'Algorithme 5) expriment deux choses différentes. Le premier est le plus évident et indique qu'un service candidat pour une composition est réputé fournir un résultat utilisable lorsqu'il est utilisé dans une composition avec un ensemble donné de valeurs d'entrées pour cette composition. Ces valeurs d'entrée ne peuvent être que l'ensemble des entrées fournies par l'utilisateur, car l'algorithme QoS choisit les services avant la phase d'exécution et ne connaît donc que les valeurs d'entrée de la composition. Ce paramètre exprime donc que, pour un ensemble donné de valeurs d'entrées pour la composition, un service a été utilisé à l'intérieur de la composition et a fourni des résultats utiles (ce paramètre est testé aux lignes 24-28).

Le second paramètre indique que, pour une composition précédente, un ensemble de services utilisés ensemble (et pas nécessairement enchaînés) ont produit des résultats utilisables. Cette indication signifie que ces services sont susceptibles de partager un contexte commun (par exemple l'observation d'un objet particulier). L'algorithme de composition parcourt le contenu des compositions précédentes ayant rempli avec succès les exigences qui leur avaient été imposées. Pour chaque composition rencontrée au cours de laquelle le service en cours d'évaluation a été interrogé conjointement à un autre service candidat à la composition courante, le poids du service en cours d'évaluation est augmenté (lignes 15-19).Ces deux paramètres sont générés automatiquement par le processus d'exécution.

La dernière partie de l'évaluation de QoS dans notre algorithme de composition est liée aux retours d'expérience des utilisateurs. Une trace de l'évaluation de la qualité d'un service est conservée, et cette évaluation est liée à chacune des sorties qu'un service peut fournir. Lorsqu'aucune évaluation de la qualité n'est disponible pour une sortie d'un service, une valeur neutre par défaut est utilisée pendant la composition pour cette évaluation. Si une évaluation de la qualité est trouvée pour un service concernant une sortie qui peut être utilisée pour la composition, cette valeur est utilisée pour la sélection de ce service dans la composition pour cette sortie (lignes 20-23).

Cette évaluation de l'utilisateur est utilisée automatiquement lors de la composition, mais elle n'est pas générée automatiquement. Si un service fournit un résultat que l'utilisateur juge valable, cet utilisateur peut utiliser un bouton dans l'interface Web pour valider le résultat. Si tel est le cas, la valeur de qualité associée à ce service pour la sortie concernée est augmentée. Les services qui font finalement partie d'une composition sont sélectionnés en fonction de leurs propriétés QoS individuelles (même si le second paramètre de l'historique d'exécution joue dans une certaine mesure le rôle de «degré de centralité» pour un service que l'on rencontre parfois dans des méhodes de compositions préexistantes). Les approches comparant des compositions entières et choisissant une composition plutôt qu'une autre (Rodriguez-Mier et al. 2016; Zhao et al. 2013) et non pas un service plutôt qu'un autre pendant la phase de sélection des compositions ne sont pas applicables dans notre cas. A titre d'exemple, il y a 262 services élus lorsqu'une composition demande une «température effective» et 150 pour «vitesse radiale héliocentrique». En explorant toutes les compositions possibles, on obtiendrait 262 * 150 = 39300

compositions possibles seulement pour ces deux sorties. L'algorithme d'évaluation de la QoS de chaque service est l'Algorithme 5.

Algorithme 5: Evaluation de la QoS

	Entrée:
1:	*U = {(I_0,V_0)…(I_n, V_n)} les couples (Entrée I, Valeur V) données par l'utilisateur*
2:	*Le fichier F issu de l'algorithme de composition*
3:	*Le fichier H_1 contenant l'historique des compositions précédentes*
4:	*Fichier H_2 contenant, pour toutes les sorties d'un service l'estimation de la qualité de la sortie considérée sous la forme (service, sortie, qualité). Cette estimation provient des*
5:	*retours des utilisateurs*
	Fichier H_3 contenant les valeurs d'entrées des compositions précédentes pour lesquelles
6:	*le service a fourni des informations utiles sous la forme (service, entrée, valeur)*
	W(H_1), W(H_2), W(H_3): Les poids respectifs des estimations de qualité données par les contenus de H_1,H_2,H_3. Ces poids sont fixes à 0.5 par défaut.
7:	**Sortie:**
	Fichier F_2 contenant les services avec leurs entrées, leurs sorties et les poids associés à
8:	*chaque sortie.*
9:	**Représentation interne:**
10:	*s= {(I_0…I_n),(O_0…O_n)}, (I_0…I_n) entrée d'un service, (O_0…O_n) sorties d'un service*
	S = {s} liste de services
	Q_s estimation de la qualité pour le service s

```
11: For each s in F:
12:     S <= S + s
13: For each s ∈ S:
14:     Qs = 0
15:     For each workflow W in H1:
16:         If s in W:
17:             For each service w ∈ W:
18:                 If w != s and w ∈ S:
19:                     Qs = Qs + (0.1 * W(H1))
20:     For each O in (O0…On):
21:         For each (service,output,quality) in H2:
22:             If service == s & output == o:
23:                 Qs = Qs + (quality * W(H2))
24:     For each (service,input,value) in H3:
25:         If service == s:
26:             For each (I,V) in U:
27:                 If I == input and value == V:
28:                     Qs = Qs + (0.1 * W(H3))
29: Print s,(I0…In),(O0…On),Qs in F2
```

Le graphe acyclique dirigé résultant de l'algorithme de composition est transformé en un graphe acyclique dirigé avec des arcs pondérés. Le poids de chaque arc est la qualité estimée pour l'information fournie (le nœud de destination de l'arc) donnée par le service (le nœud source de l'arc). Un exemple d'un tel graphe dirigé est donné à la Figure 35. Les rectangles bleus indiquent les services élus dans le workflow, les ovales bleus les informations utilisées par les services. Les informations sans arcs

entrants sont données par l'utilisateur et les ovales verts indiquent des informations de sortie de la composition demandées par l'utilisateur. Les étiquettes «Qual: x» indiquent la qualité des informations fournies. Cette évaluation de la qualité commence à 0, sans limite supérieure.

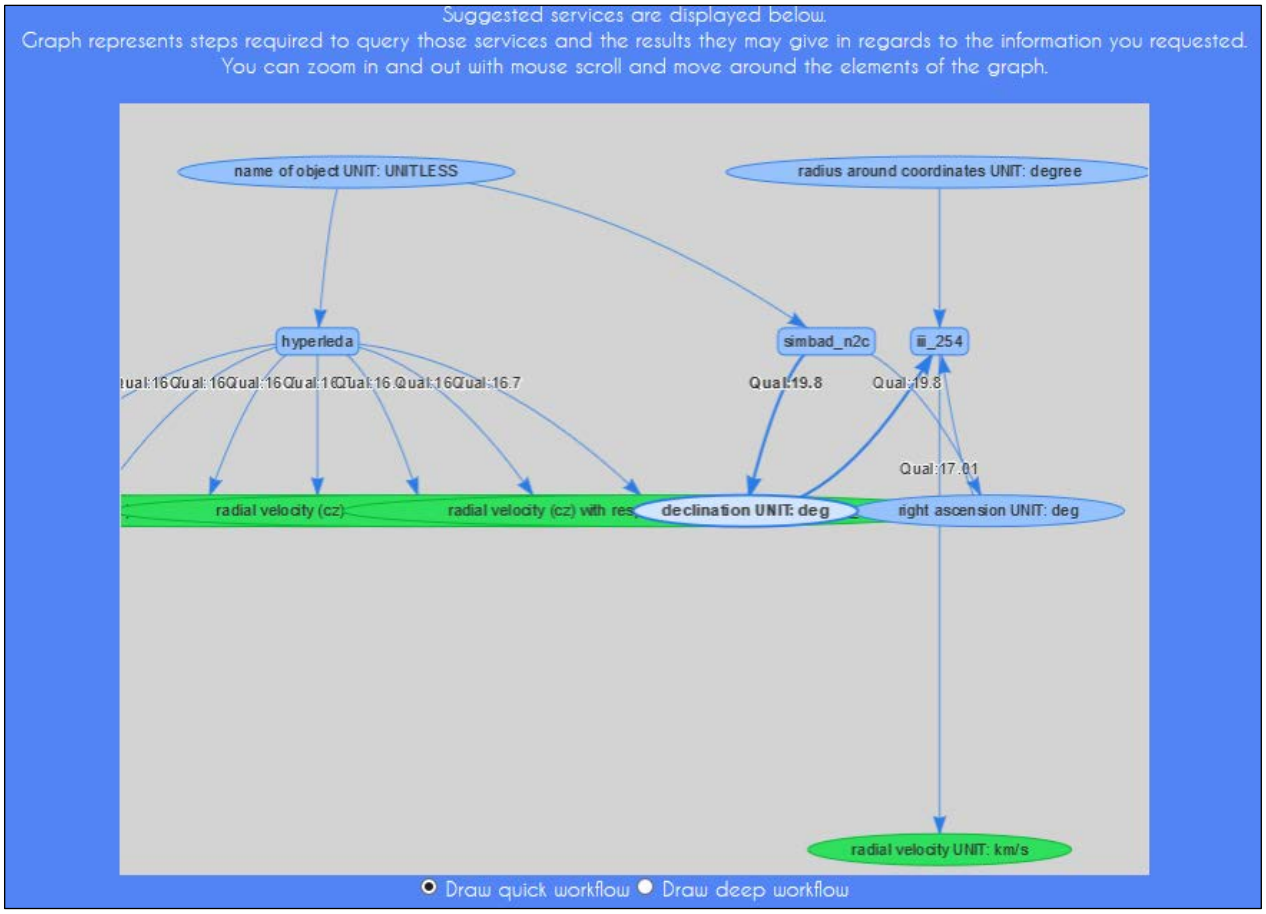


Figure 35: Extrait d'une composition

4.5.4. Phase d'exécution du workflow

Le workflow résultant de la phase de composition est un graphe orienté, dont les nœuds sont soit des informations, soit des services. A partir des nœuds sans prédécesseurs, qui sont à la première itération de l'algorithme d'exécution les informations d'entrée de la composition, l'algorithme de composition invoque les services éligibles. Les informations en sortie de ces services invoqués deviennent à leur tour des nœuds sans prédécesseur, utilisés par l'itération suivante. L'algorithme se poursuit jusqu'à ce que les nœuds sans prédécesseur soient également sans successeurs, ce qui indique les informations finales que l'algorithme d'exécution doit retourner à l'utilisateur.

Sur la base des valeurs calculées par l'algorithme d'évaluation de la QoS pour chacune des informations fournies par chaque service, l'algorithme d'exécution choisit le meilleur service pour chaque sortie demandée à la composition. Cela peut rendre certains services inutiles. C'est notamment le cas lorsqu'une sous-chaîne de la composition conduit à l'utilisation d'un service donné, et que ce service

donné n'est le meilleur service pour aucune de ses sorties. Alors, certains services dans la sous-chaîne peuvent devenir inutiles. En conséquence, une des premières actions de l'algorithme d'exécution est d'invoquer l'algorithme qui élimine les services inutiles, qui est l'Algorithme 6.

Algorithme 6: Suppression des services inutiles EliminateUselessServices

1:	**Input :** *S = {s} liste de services*
2:	**Output :** *S = {s} liste de services expurgée de tout service inutile*
3:	**Représentation interne:** *s= {(I_0...I_n),(O_0...O_n)}, (I_0...I_n) entrées d'un service, (O_0...O_n) sorties d'un service*
4:	Cflag = 1
5:	While Cflag == 1:
6:	Cflag = 0
7:	For each s ∈ S:
8:	Pflag = 1
9:	For each O ∈ (O_0…O_n)
10:	For each s in S:
11:	For each I ∈ (I_0…I_n)
12:	If O == I:
13:	Pflag = 0
14:	Break
15:	If Pflag == 0:
16:	Break
17:	If Pflag == 0:
18:	Break
19:	If Pflag == 1:
20:	S <= S - s
21:	Cflag = 1
23:	Return S

L'algorithme d'exécution est l'Algorithme 7.

Algorithme 7: Exécution du workflow

	Entrée :
1:	*Fichier F_2 issu de l'algorithme de QoS*
2:	*Fichier F_p produit par l'algorithme de composition contenant les paramètres du service (URL, QuerySoftware)*
3:	*G = {(I_0,U_0,F_0,V_0)…(I_x,U_x,F_x,V_x)} Individu, Unité, Format, Valeur: informations fournies par l'utilisateur*
	Sortie :
	Résultats produits par l'exécution des services, liste « Result »
	Représentation interne:
4:	*C = {s* s,$(I_0…I_n)$,$(O_0…O_n)$*} liste des services à appeler pour l'exécution avec leurs entrées et*
5:	*sorties*
	AI = {(I_0,U_0,F_0,V_0)…(I_x,U_x,F_x,V_x)} Individu, Unité, Format, Valeur: entrées des services disponibles pendant la composition
6:	For each s,$(I_0…I_n)$,$(O_0…O_n)$,Q_s in F_2:
7:	C <= C + s,$(I_0…I_n)$,$(O_0…O_n)$
8:	EliminateUselessServices(C)
9:	For (I,U,F,V) in G:
10:	AI <= AI + (I,V)
11:	While C = ! {0}
12:	For s,$(I_0…I_n)$,$(O_0…O_n)$ ∈ C
13:	If $(I_0…I_n)$ in AI:
14:	For each I_x:
15:	V_x = GetValue(AI,I_x)
16:	url = GetUrl (F_p, s)
17:	querysoft = GetQuerySoftware(F_p,s)
18:	Result = RunService(s,url,querysoft, $(I_0…I_n)$, $(V_0…V_n)$)
19:	Write(H3, $V_0…V_n$,s)
20:	AI <= AI + $(O_0…O_n)$
21:	C <= C - s,$(I_0…I_n)$,$(O_0…O_n)$
22:	Write(H1,s)
23:	Return Result

La liste « Result » renvoyée par l'algorithme d'exécution contient chaque sortie de la composition ainsi que la valeur renvoyée pour cette sortie pendant la phase d'exécution.

Après la phase d'exécution, il est possible de modifier les valeurs des poids données à l'algorithme de QoS pour les paramètres W (H1), W (H2), W (H3). Une nouvelle évaluation de la QoS et une nouvelle exécution peuvent alors être demandées, sans passer par la phase de composition une nouvelle fois.

L'interface Web fournit également la possibilité de demander l'exécution de chaque composition disponible. L'algorithme d'évaluation de la QoS n'est pas relancé dans ce cas, l'interface appelle alors directement la phase d'exécution à partir du fichier F résultant de la phase de composition. Cela

interroge tous les services élus au cours de la composition en contournant chaque évaluation de la QoS ; pour constituer en fait une «solution d'urgence» pour les compositions dans lesquelles très peu de services contiennent effectivement les sorties requises dans un contexte de fonctionnement donné. Plus de détails à ce sujet seront donnés dans le chapitre 5.

4.6. Conclusion

Nous avons présenté dans ce chapitre un processus de composition de services basé sur l'utilisation d'une ontologie de services. Cette composition prend en compte les spécificités des services décrits dans l'ontologie. Le processus proposé est indépendant des technologies et des langages habituels utilisés dans les sciences de l'information, tels que WSDL ou UDDI. Afin de proposer le meilleur résultat possible, il prend en compte les incertitudes liées à la nature des services, et s'appuie sur l'historique des compositions précédentes.

Le Tableau 14 résume différents critères concernant les approches existantes et la composition proposée dans ce manuscrit. Ces critères sont ceux retenus dans une étude sur la composition de services Web parue en 2011 (Bartalos et al. 2011). Le tableau est issu de cette étude mises à part les cinq dernières approches indiquées, ajoutées pour bénéficier de comparaisons plus récentes. Le contrôle sur le flot de données indique si l'utilisateur peut modifier le flot de données ou de services dans la composition, l'extension fonctionnelle indique si des critères non liés aux entrées et aux sorties des services sont pris en compte dans la composition (comme des conditions préalables de fonctionnement). Le tableau signale si les approches quantifient la qualité des compositions (QoS) ou non, si un plan optimal tenant compte de cette qualité est construit. L'évaluation des performances (en temps, en qualité, etc.) est également indiquée.

Tableau 14: Comparaison entre les approches de composition existantes et la composition proposée dans ce manuscrit

Approche	Contrôle sur le flot de données	Extension fonctionnelle	QoS	Plan optimal	Evaluation des performances
Bartalos et al. [9,8]	Yes	Yes	Yes	Yes	Yes
Huang et al. [23,25]	Yes	No	Yes	Yes	Yes
Alrifai et al. [3]	No	No	Yes	No	Yes
Rosenberg et al. [48]	No	No	Yes	No	Yes
Lécué et al. [34]	No	No	Yes	No	Yes
Shin et al. [49]	Yes	Yes	No	-	Yes
Gamha et al. [20]	Yes	No	No	-	No

Klusch et al. [27,28]	Yes	Yes	No	-	Yes
Sirin et al. [49]	HTN	Yes	No	-	No
Kona el al. [31,30]	Yes	Yes	No	-	Yes
M. Lin et al. [37]	No	No	Yes	Yes	Yes
Agarwal et al. [1]	No	No	Yes	Yes	No
N.Lin et al. [38]	HTN	Yes	No	-	No
Karakoc et al. [26]	No	Yes	No	-	Yes
Li et al. [36]	HTN	Yes	No	-	Yes
Bansal et al. 2014	No	Yes	Yes	Yes	No
Li et al. 2015	No	No	No	No	No
Rodriguez-Mier et al. 2015	No	Yes	Yes	Yes	No
Puttonen et al. 2015	No	No	No	Yes	Yes
Approche proposée	Yes	Yes	Yes	Yes	No

L'approche proposée dans ce manuscrit autorise le contrôle du flot de données par le choix des poids accordés à chacun des paramètres de QoS, puis propose un plan optimal suivant ces poids. L'évaluation des performances des compositions n'est pas proposée.

Chapitre 5 : Application

5.1. Introduction

Ce chapitre présente l'orchestration, dans l'application CASAS (*Composing Automatically and Semantically Astrophysical Services*) des différentes briques développées précédemment. Cette application vise à fournir une réponse aux difficultés exprimées dans le contexte astrophysique. Elle se propose d'utiliser des services hétérogènes qui sont aussi bien des services Web s'éloignant des standards habituellement rencontrés dans les sciences de l'information, que des services locaux au serveur hébergeant CASAS. Ces services peuvent provenir de l'IVOA, ou bien être indépendants de toute structure fédérée existante.

Après avoir décrit le principe général et l'architecture de CASAS, nous illustrerons son fonctionnement sur un exemple concret. En suivant et avant de conclure, sera présentée l'utilisation sur le même exemple de l'application Taverna, la plus employée actuellement dans ce contexte astrophysique, afin de les comparer et de discuter de ses apports et de ses limites.

5.2. Principe général d'implémentation

Le principe général de l'application est simple. Il s'articule autour de 3 modules : une ontologie de services, un moteur de composition et d'orchestration et une interface utilisateur, présentés sur la Figure 36.

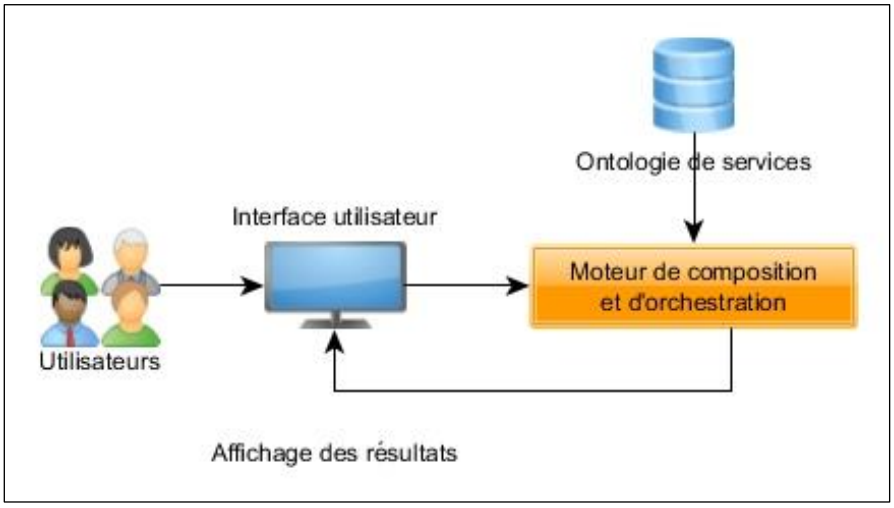


Figure 36 : Architecture générale de l'application

- Le module « Ontologie de services». S'appuyant sur les résultats présentés aux chapitres 2 et 3, ce module s'appuie sur deux composantes : une composante décrivant les éléments des services indépendants du domaine d'application, et une composante décrivant les éléments dépendants de ce domaine.
- Le module de « Composition et d'orchestration». Ce module est composé d'un mécanisme de requête indépendant du domaine. Il permet de reconnaître le langage « naturel » utilisé par l'utilisateur, et établit une correspondance entre le contenu de la requête avec les

concepts et les individus exprimés dans l'ontologie. Présenté au chapitre 4, il contient un système de raisonnement capable d'utiliser le contenu de l'ontologie pour établir les compositions de services à même de répondre à la requête exprimée.

- Le module « interface utilisateur ». Ce module facilite l'expression des requêtes et affiche le résultat des compositions effectuées.

5.3. L'architecture de CASAS

L'architecture de CASAS (Figure 37) présente, de manière plus détaillée, la mise en œuvre du principe général dans le cas de l'astrophysique.

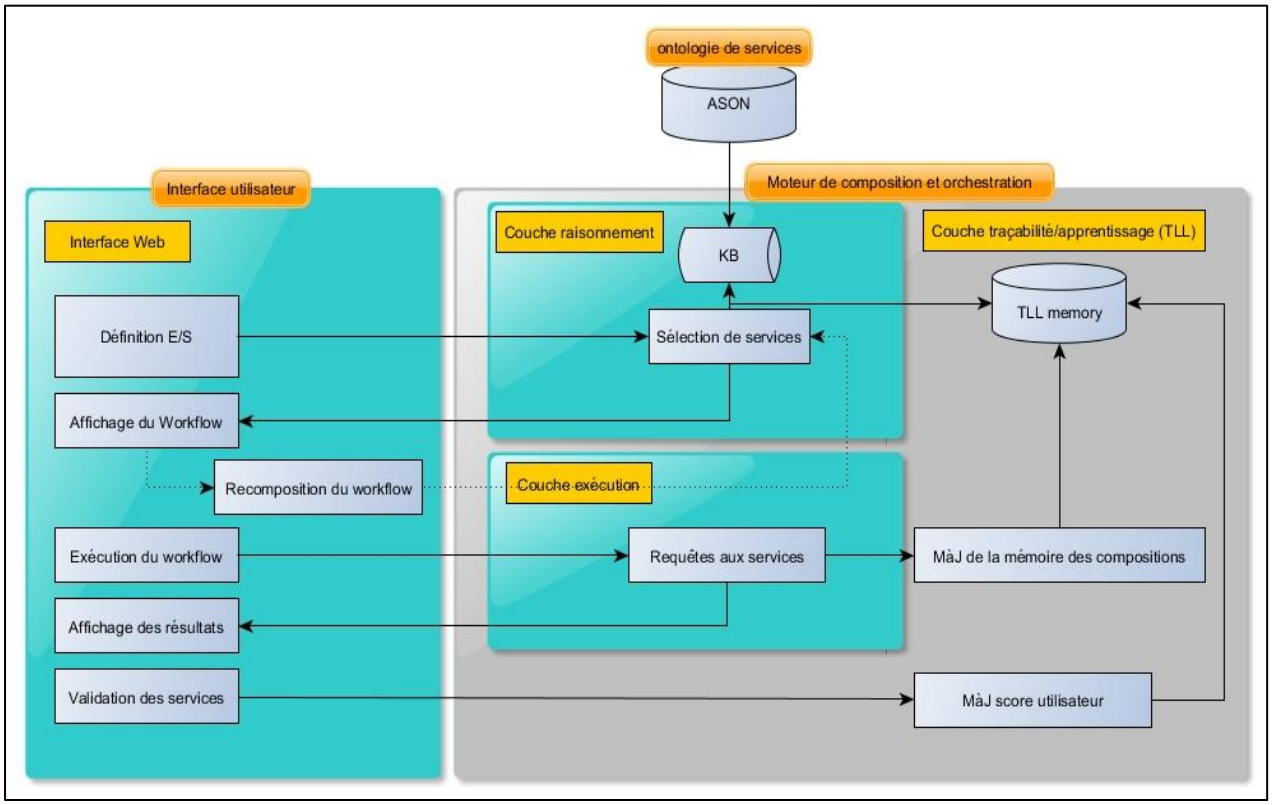


Figure 37: Architecture interne de CASAS

5.3.1. L'ontologie de services

Dans ce contexte, l'ontologie de services se nomme ASON (*Astrophysical Services ONtology*). L'ontologie de services ASON contient tous les services disponibles pour la composition, et de nouveaux services peuvent être enregistrés. L'enregistrement d'un service dans cette ontologie nécessite un fichier XML contenant une description générale du service, et une description de chaque sortie et de chaque entrée avec les unités et les formats associés. Chaque description est exprimée en langage naturel, et automatiquement associée à l'individu le plus pertinent dans l'ontologie. Les deux dernières informations dans ce fichier XML sont L'URL du service et son protocole, s'il existe un

protocole défini dans ASON auquel le service se rattache (par exemple, pour un nouveau service SSAP de l'IVOA).

Ces informations d'URL et de protocole ne sont pas suffisantes, lorsque le service à intégrer ne s'appuie pas sur un protocole décrit dans ASON. C'est notamment le cas des logiciels tels qu'Aladin, installés sur le même serveur que celui qui héberge CASAS. Il est alors nécessaire d'encapsuler le fonctionnement du logiciel via l'utilisation d'un script. Ce script doit avoir les entrées du service comme paramètres d'entrée, et retourner les sorties du service pour que son appel puisse être intégré dans l'algorithme de composition. Il est rattaché à l'ontologie via le concept de « QuerySoftware » présenté au chapitre 2.

Les services OV disponibles dans CASAS ont été automatiquement enregistrés en utilisant la méthode présentée au chapitre 3. Leurs descriptions viennent en grande majorité de l'interrogation de l'annuaire VO-Paris[53]. Les services non-OV ont été enregistrés manuellement, en générant un fichier XML par service. Pour illustrer les capacités de CASAS, les services non-OV suivants ont été intégrés :

- Un service interrogeant la version non-OV de la base de données Hyperleda
- Un script faisant appel à une version d'Aladin locale au serveur de CASAS
- L'utilisation d'AladinLite par le Web
- Un service proposant des mesures de champ magnétique sur les étoiles

La sélection des services est basée sur une correspondance syntaxique avancée entre une description utilisateur des résultats attendus du workflow et le contenu sémantique des sorties de services disponible dans CASAS. Cette méthode résout les problèmes de non-conformité des mots-clés («color index» et «colour index» vont correspondre), mais ce n'est pas son principal avantage.

Le principal avantage de cette correspondance syntaxique est de pouvoir décrire les sorties du workflow sans modifier le comportement habituel de l'utilisateur. Tout niveau de détail est acceptable, et voici un exemple: la « heliocentric radial velocity » donnera une sélection de services plus générique que la « heliocentric radial velocity in optics », qui conduira elle-même à la même composition que « heliocentric radial velocity, optical measurement ». Les trois descriptions seront identifiées dans CASAS et guideront l'algorithme de composition vers le niveau de détail approprié.

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

5.3.2. Moteur de composition et d'orchestration

CASAS résout toutes les contraintes de la composition en une seule exécution, de sorte que chaque service susceptible de fournir n'importe quelle information demandée soit sélectionné en un seul appel à l'algorithme de composition.

Le rôle de la couche de raisonnement est de trouver la bonne composition des services. Cette couche est chargée de sélectionner les enchaînements de services capables de produire les sorties du workflow. Tous les services utiles à la composition sont sélectionnés dans une première itération de l'algorithme de composition. Lorsque des informations utiles à la composition peuvent être fournies par plusieurs services, il est nécessaire de faire un choix. Suivant la méthode décrite au chapitre 4, l'algorithme de composition calcule alors des pondérations pour chaque service, afin de définir leur priorité respective. Ces pondérations sont dynamiques et dépendent du contexte dans lequel le service doit être utilisé. Elles sont calculées en utilisant les informations fournies par la couche de traçabilité / apprentissage (*Traceability Learning Layer*, TLL) décrite ci-après, et le meilleur service après calcul est élu dans le premier workflow proposé à l'utilisateur («*quick workflow*») pour l'information donnée.

Une fois le workflow composé, il est nécessaire de faire appel aux services. Cela comprend la phase d'orchestration (qui détermine l'ordre dans lequel les services doivent être invoqués) et la phase d'exécution (qui invoque les services) (Puttonen et al. 2013) . La couche d'exécution est chargée de la mise en œuvre de ces deux phases. La couche d'exécution est très sensible au contenu réel des services. Un service est enregistré en tant que fournisseur d'une information même s'il ne la fournit effectivement que pour un seul cas d'utilisation. Par exemple, le service «j_apj_693_1084» est enregistré dans ASON comme fournisseur de mesure de «vitesse radiale», mais sollicité pour la galaxie spécifique «MRK1224», elle ne fournira pas cette mesure. Ces «trous d'exécution» sont réduits par la couche de traçabilité / apprentissage, qui garde la mémoire des combinaisons de services qui ont fonctionné ensemble avec succès et les contextes dans lesquels les requêtes de services ont réussi.

Le TLL est la dernière composante de CASAS. Elle fournit à la couche de raisonnement des informations sur les exécutions antérieures. Lorsque l'algorithme de composition rencontre un service, il interroge le TLL sur:

- (1) L'historique des contextes d'utilisation des services (avec les mêmes valeurs d'entrée)
- (2) Le nombre de services candidats pour la composition en cours d'élaboration qui ont été précédemment utilisés en conjonction avec ledit service et qui ont rempli leurs exigences
- (3) La qualité du service pour les informations demandées

Le TLL conserve:

- Les paramètres d'entrée d'un service lorsque ce service a fourni les informations qui lui ont été demandées lors d'une requête
- La composition des workflows qui ont satisfait leurs exigences
- L'estimation de la qualité des services pour chacune de leurs informations de sortie

Le rôle exact de ces informations, enregistrées par le TLL lors de l'exécution du workflow et utilisées lors de la composition, demande quelques précisions. Une fois qu'un workflow possible a été défini, il est susceptible de contenir des dizaines de services concurrents pour des informations très courantes. Pour chaque service, l'algorithme de composition calcule alors une estimation de la QoS. Comme indiqué au chapitre 4, cette estimation est fonction de l'importance de chacune des informations (1),(2) et (3) exposées précédemment sur la composition finale, et de la valeur associée à cette information. L'importance associée à chacune des informations est traduite par un poids associé à cette information. Les poids associés aux informations du TLL ne sont pas fixés une fois pour toutes, mais dépendent du contexte dans lequel le service est utilisé et peuvent être fixés par l'utilisateur.

Ces poids aident à résoudre un des biais relatifs au choix des services qui pourrait s'installer lors de l'utilisation de CASAS. Il est ainsi probable que des services très importants possédant de nombreuses capacités participent à plus de workflows que des services plus restreints et plus spécialisés. Cela causera une augmentation des valeurs liées à l'historique des compositions en faveur des grands services. Cette augmentation pourrait amener à une domination de ces grands services au détriment de services plus spécialisés, qui pourraient pourtant mieux correspondre à certains cas d'utilisation bien définis. Ce type de biais est contré par le poids portant sur la spécialisation des services (information (3) du TLL), qui améliore la sélection de catalogues plus petits et plus spécialisés par rapport à des grands catalogues génériques. L'importance de chacun des poids associés aux informations du TLL peut varier d'un cas à l'autre, et il est possible d'utiliser l'interface pour ajuster ces paramètres au cas par cas. Une modification de ces poids entraînera la composition d'un autre workflow.

Par conséquent, si l'utilisateur souhaite donner la priorité à l'information exprimant la spécialisation des services, cela peut être fait en augmentant le poids associé à l'information (3) du TLL et en diminuant les valeurs des poids associés à (1) et (2) dans la composition. En outre, si l'utilisateur veut tester chaque combinaison possible de services, cela peut être fait en utilisant le «*deep workflow*». Ce «flux de travail profond» interroge tous les services disponibles, donc il n'est sous l'influence d'aucun biais découlant de l'histoire des compositions précédentes.

Un autre type de biais peut apparaître, lié à la connaissance préexistante de l'utilisateur quant à l'existence et au contenu des services. Un utilisateur fera plus souvent appel à des services qu'il connaît a priori, plutôt qu'à des services découverts après analyse de la vaste gamme disponible. Ce biais est exclu de CASAS, qui élabore ses compositions en raisonnant sur tous les services enregistrés dans le système. Néanmoins, des biais que CASAS ne traite pas pourraient apparaître lors de l'utilisation du système, et il est également possible que des biais imprévus et causés par le raisonnement automatique sous-jacent à l'approche soient découverts.

5.3.3. L'interface utilisateur

L'interface utilisateur est implémentée sous la forme d'une interface Web[54]. Elle permet aux utilisateurs d'accéder à la composition automatique de services. L'utilisation de cette interface est simple, l'utilisateur indique uniquement les données qu'il peut fournir et les informations de sortie qu'il souhaite obtenir. Dans cette première version de l'interface les seules entrées possibles sont le nom d'un objet et un rayon autour de cet objet, bien que l'algorithme de composition soit susceptible de prendre toute information décrite dans l'ontologie comme valeur d'entrée. La composition de services résultante est affichée sous la forme d'un workflow. L'interface permet deux modes de composition de services : un mode général (« deep workflow ») qui comprend l'intégralité des possibilités de services offertes indépendamment de leur classement vis-à-vis des mesures de QoS ; et un mode par défaut («quick workflow») qui est un sous-ensemble du cas précédent, plus rapide, composé à l'aide du TLL permettant de choisir les meilleurs services disponibles.

Pour les besoins du « quick workflow », l'interface permet également de modifier le comportement de l'algorithme de composition en augmentant ou en diminuant l'influence de chacun des paramètres de QoS. Dans les deux modes, les résultats affichés sont les quantités astrophysiques demandées par l'utilisateur.

5.4. Cas d'utilisation

Nous avons testé les méthodologies proposées sur quatre cas d'utilisation réels. Ces quatre cas d'utilisation ont été définis pour illustrer les capacités de l'approche à utiliser des services hétérogènes, et parce que leurs résultats sont des quantités astrophysiques (températures, vitesses radiales…) qui peuvent être comparées avec des valeurs disponibles dans des bases de référence. Ces cas d'utilisation sont accessibles par l'interface Web, les données nécessaires pour les utiliser sont préenregistrées.

Dans le premier cas, il s'agit de rechercher une carte de champ (c'est-à-dire une carte des objets présents dans une partie du ciel), accompagnée d'une carte du ciel interactive autour de l'étoile SIRIUS. La vitesse radiale héliocentrique de l'étoile et sa température sont aussi demandées. Cet exemple illustre l'utilisation de services OV (pour la température et la vitesse) en même temps qu'un service Web non OV (AladinLite, pour la carte du ciel interactive) et d'un service local, Aladin pour la carte de champ.

Le second cas d'utilisation fait appel à un fournisseur de données non-OV (pour la détection de champ magnétique) conjointement avec des services OV pour trois autres mesures portant sur l'étoile « RR Lyrae ».

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

Le troisième cas recherche des spectres dans différentes longueurs d'onde (Infra-rouge, ultraviolet…) pour la galaxie « M1 ». Ce cas illustre comment des spectres peuvent être obtenus, en tenant compte de la longueur d'onde observée lors de la sélection des services adéquats. Il interroge pour cela des services répondant au protocole SSAP de l'IVOA.

Le dernier cas d'utilisation, intitulé «Galactic example 2», recherche des mesures de magnitude, de vitesse et d'extinction concernant la galaxie « MRK1224 ». Ce cas fait appel à une base de données spécifiquement dédiée à HyperLeda non intégrée dans l'OV.

Dans ces quatre cas, les résultats obtenus sont conformes aux attentes et peuvent être comparés aux données enregistrées dans des bases de référence telles que Simbad@CDS, lorsque les données recherchées y figurent. Le dernier cas fait l'objet d'une présentation plus détaillée dans les sections suivantes, d'abord par l'intermédiaire de l'approche CASAS puis par l'utilisation de l'approche Taverna.

5.4.1. Résolution à l'aide de l'approche CASAS

La première étape consiste à définir les valeurs d'entrée et de sortie pour la composition. Cette définition est illustrée en Figure 38, en utilisant le cas d'utilisation 4.

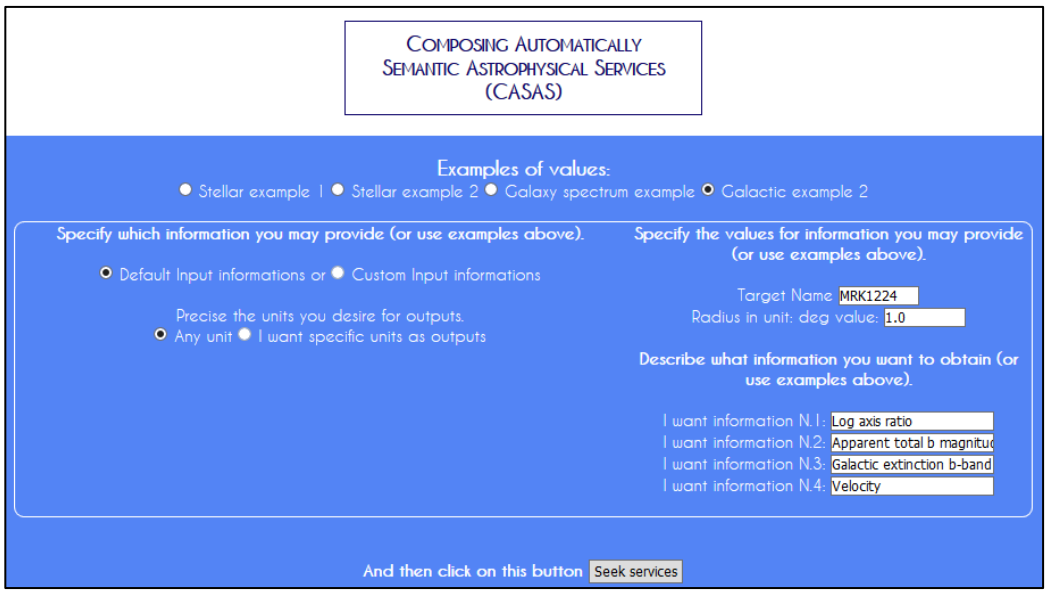


Figure 38: Définition des paramètres d'entrée et de sortie dans CASAS

Comme on peut le constater, les données renseignées sont, pour les entrées le nom d'une cible (la galaxie « MRK 1224 ») et l'incertitude sur ses mesures de coordonnées (Radius, 1 degré). Des mesures, telles que la magnitude (Apparent total b magnitude) ou la vitesse (Velocity) sont attendues comme résultat. Pour ce cas particulier, les entrées et les sorties recherchées sont préenregistrées dans l'interface. Le bouton «Seek services» lancera la composition du workflow, et le workflow obtenu est présenté à la Figure 39. Dans ce dernier, les rectangles bleus représentent les services élus, les ovales bleus sont les entrées utilisées par ces services. Les informations sans arcs entrants sont les entrées fournies par l'utilisateur et les ovales verts représentent les informations souhaitées par l'utilisateur.

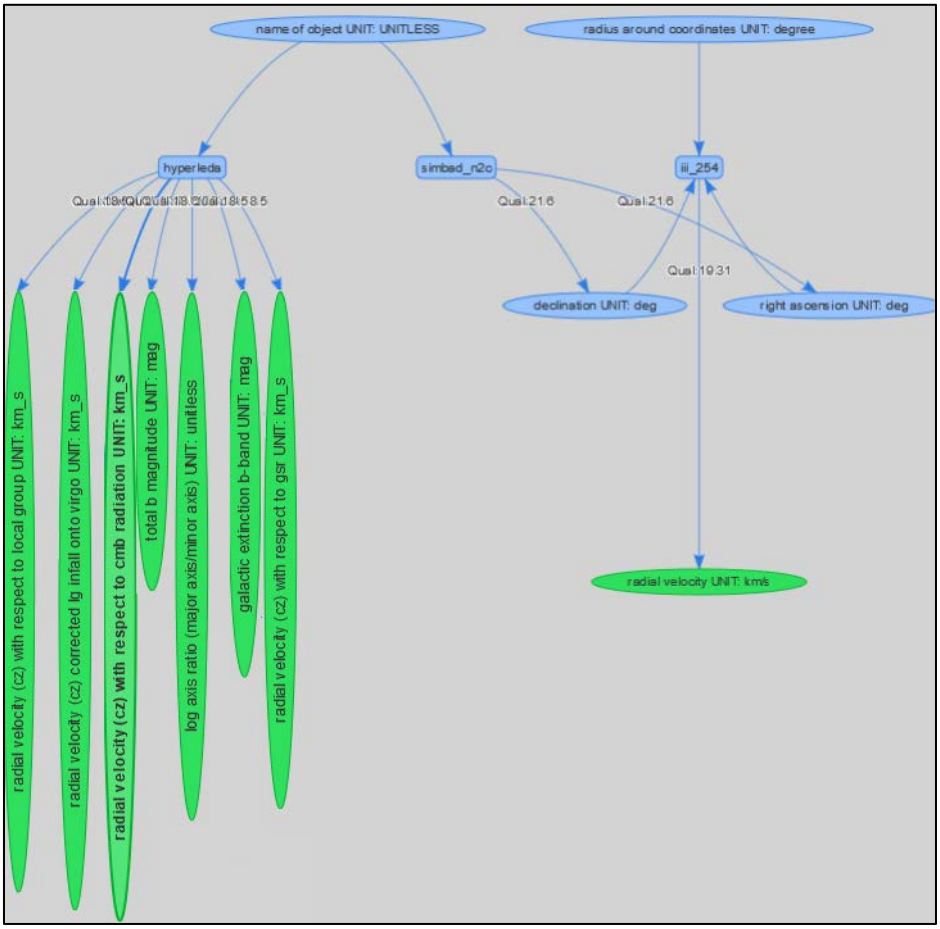


Figure 39: Présentation d'un workflow dans CASAS

La Figure 40 présente la partie de l'interface concernant l'éditeur permettant à l'utilisateur de modifier les poids accordés aux différents paramètres de QoS définis au chapitre 4 et dont l'objectif est de modifier le comportement de l'algorithme de composition. Ces modifications sont prises en compte lors des nouvelles compositions, par la sollicitation sur l'interface du bouton (« Recompose quick workflow »).

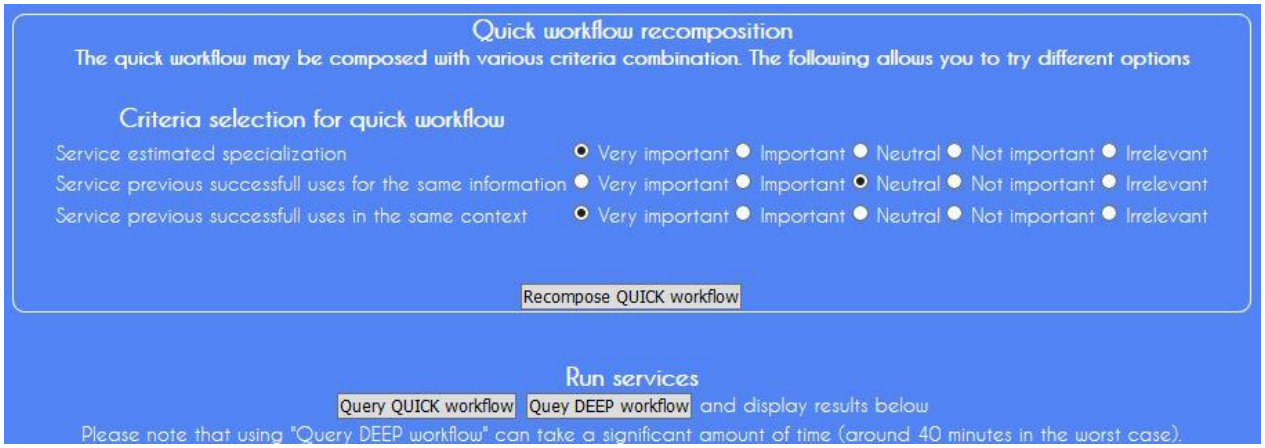


Figure 40: Redéfinition des poids et exécution des workflows

La dernière étape concerne l'affichage des résultats, suite à l'exécution du workflow choisi. Ces derniers sont affichés avec une indication des services qui ont permis de les obtenir. Les résultats pour le cas illustré sont exposés à la Figure 41.

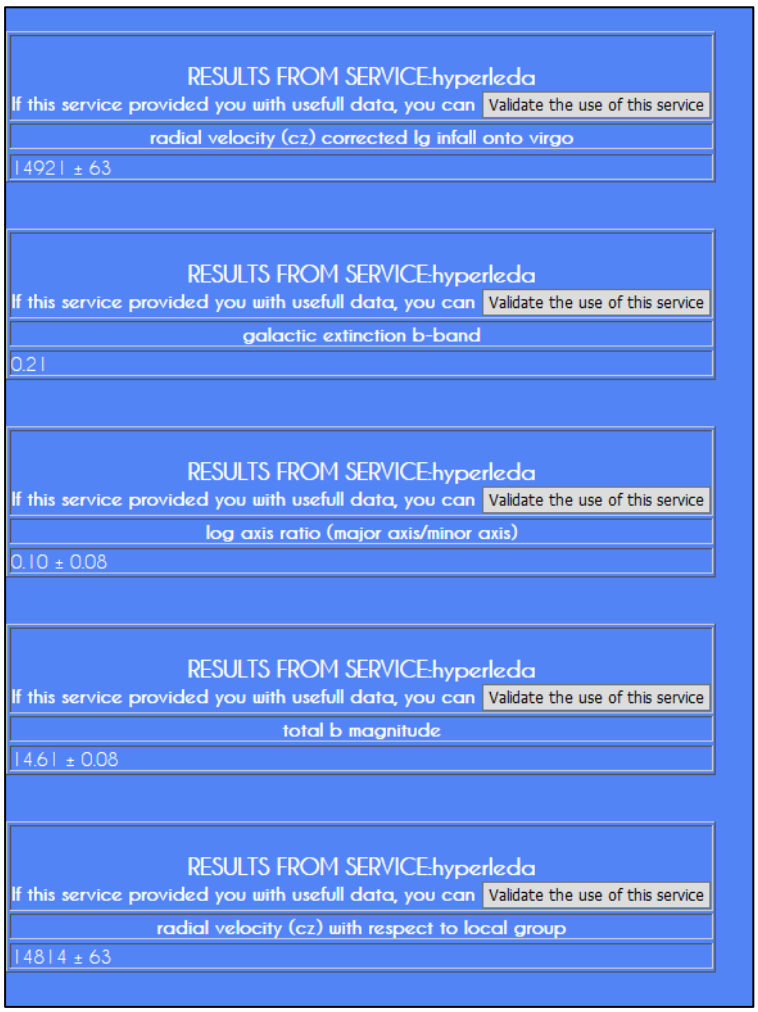


Figure 41: Résultats affichés dans CASAS

Compte tenu des informations demandées par l'utilisateur, on obtient pour la magnitude « total b magnitude », la valeur de 14.61 + /- 0.08, et deux valeurs pour la vélocité (« radial velocity (cz) corrected lg infall onto virgo », de valeur 14921 +/- 63 et «radial velocity (cz) with respect to local group » de valeur 14814 +/- 63). Les deux valeurs liées à la vélocité viennent du fait que cette notion est peu précise, et que l'instance Hyperleda du concept « Services » est lié dans l'ontologie à plusieurs instances du concept « velocity ».

Le bouton « validate this service » donne au TLL un retour d'expérience. C'est ce retour de l'utilisateur qui indique que le service associé, avec ses entrées associées, a fourni des informations utiles pour le workflow.

5.4.2. Résolution à l'aide de l'approche Taverna

Afin de montrer l'apport de CASAS, cette section présente le processus de résolution du quatrième cas d'utilisation par le système Taverna, qui est le logiciel le plus avancé en astrophysique pour la composition de services. Le gestionnaire de workflows Taverna est un projet de l'incubateur d'Apache. Il permet d'automatiser l'exécution de workflows dans plusieurs domaines, dont l'astrophysique. Cependant, la définition de workflows dans Taverna n'est pas automatisée et demande une certaine connaissance technique.

Comme dans l'approche CASAS, la première étape consiste à élaborer un workflow. Cette étape peut être facilitée par l'existence d'un workflow équivalent dans une base de workflows disponibles sur le site Web MyExperiment[55], issus de retours d'expériences. La recherche de workflows se fait à l'aide de l'interface présentée à la

Figure 42, par l'intermédiaire de mots-clés. Les workflows disponibles sont classés suivant différents critères tels que les catégories (type) ou les annotations (tags). Les workflows déjà composés et leur description, conditionnent donc la faisabilité et le niveau de correspondance entre les exigences d'un cas d'utilisation et l'utilisation de Taverna pour les satisfaire. Puisqu'il est peu probable qu'un workflow disponible convienne parfaitement aux exigences d'un utilisateur, les possibilités de modification peuvent également constituer un critère de choix important. Dans le cas où aucun workflow ne convient, l'utilisateur doit faire appel à ses compétences propres pour le définir, le tester et éventuellement le partager dans la base de retours d'expériences, ce qui n'est pas aisé. Pour le cas d'utilisation concerné (« Galactic example 2 »), la recherche a été facilitée par la présence d'un workflow préexistant. Le workflow sélectionné est affiché, accompagné de son titre et de sa description (Figure 43). L'exécution du workflow peut alors être réalisée.

[55] https://www.myexperiment.org/

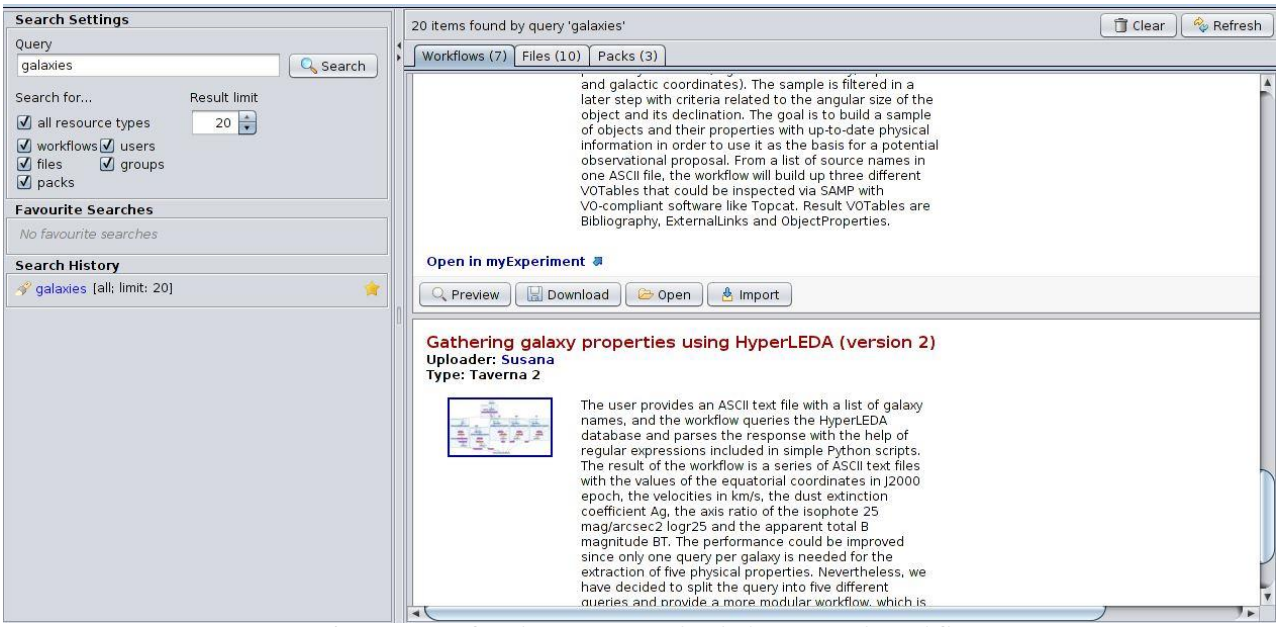
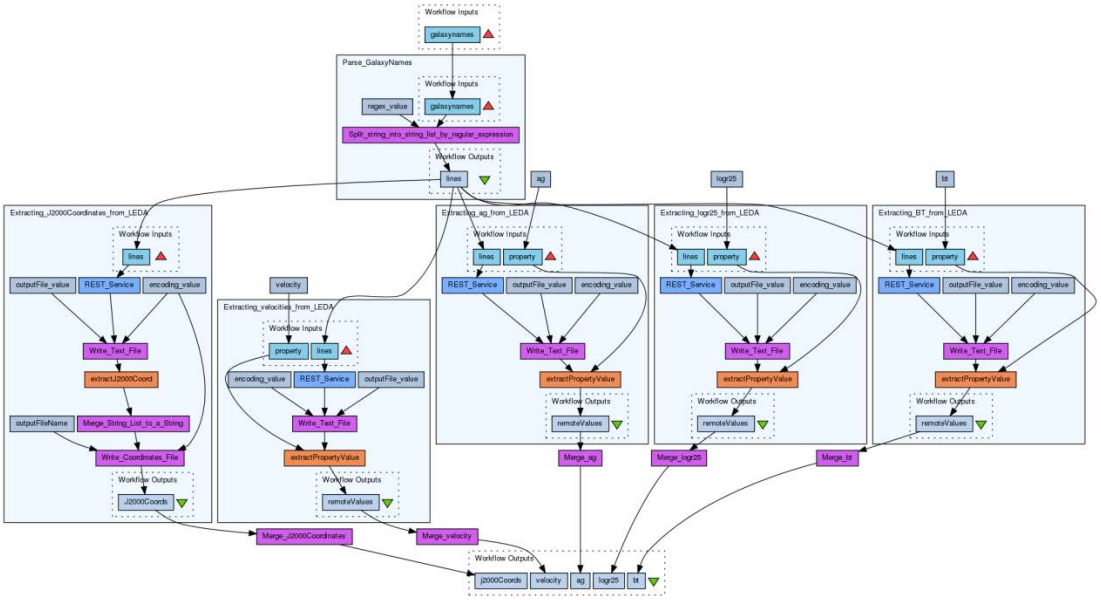


Figure 42: Interface de Taverna pour le téléchargement de workflows

Figure 43: Composants du workflow «Gathering galaxy properties using HyperLEDA»



Les résultats obtenus sont les mêmes que ceux obtenus par l'approche CASAS.

.

5.5. Discussion : CASAS et Taverna

La similitude des résultats obtenus entre les deux approches pose le problème de l'intérêt du travail accompli sur l'approche CASAS. Aussi, l'objectif de cette section est de comparer les deux approches suivant les 13 critères définis par les auteurs Lagares Lemos et al. (Lemos et al. 2016). Dans cette revue, les auteurs proposent un cadre pour la caractérisation de différentes approches de composition de services Web, et notamment de Taverna. Nous proposons d'utiliser ce même cadre pour caractériser l'approche CASAS. Les tableaux 13, 14 et 15 présentent l'étude comparative des deux approches, étant entendu que la caractérisation de Taverna provient de l'article (Lemos et al. 2016).

Concernant le critère du «Langage», présenté dans le Tableau 15, les différences les plus significatives sont que Taverna s'appuie sur des technologies très utilisées (SOAP/REST, JSON/XML…), alors que CASAS utilise une technologie adaptée aux spécificités du domaine astrophysique basée sur la technologie OWL-S.

Concernant le critère «utilisateur cible», le profil utilisateur ciblé est également différent. Taverna cible les utilisateurs finaux possédant des compétences techniques, alors que CASAS s'adresse à des utilisateurs novices et sans aucune connaissance technique. CASAS peut également utiliser tout service enregistré dans ASON, et ne se limite pas aux services Web.

D'autres différences, moins importantes, existent en particulier dans la gestion des flux de données. Alors que Taverna s'appuie sur des briques logicielles dédiées pour la gestion des unités dans lesquelles les quantités sont exprimées, dans CASAS cette gestion se fait au moyen de services décrits de la même façon que tout autre service dans ASON.

Langage							
				Eléments de la composition			
	Composants	**Application visée**	**Notation, paradigme**	**Contrôle du flux**	**Flux et transfert de données**	**Gestion des erreurs**	**Utilisateur visé**
Taverna	Compose avec les données. Formats JAVA, JSON et XML. Supporte les protocoles SOAP et REST. Interactions de type "Push". Sélection des services au moment de la définition du workflow	Workflows scientifiques	Notation visuelle, diagrammes de flux. Paradigme basé sur les flux	Contrôle simple (séquence et choix alternatifs)	Flux de données pour l'échange de données, dispositifs pré-construits pour l'échange de données.Transformation de données par l'utilisation de langages de transformation.	Exceptions (ré-essais, tâches alternatives)	Utilisateur final programmeur (experts du domaine)
CASAS	Compose avec les données. Descriptions basées sur OWL-S. Supporte nativement les protocoles OV et REST. Supporte tout protocole si le logiciel d'utilisation est fourni. Interaction de type "Pull". Sélection des services au moment de la définition et du déploiement du workflow.	Workflows scientifiques, aggrégation de ressources Web.	Notation visuelle, diagrammes de flux. Paradigme basé sur les flux. Langage naturel contrôlé	Contrôle simple (séquence et choix alternatifs)	Stockage centralisé et partagé des données. Services de transformation de données exprimés indifféremment des services de récupération ou de traitement de données.	Exceptions (appels alternatifs).	Utilisateur final sans compétence technique requise

Les critères réutilisabilité et automatisation sont présentés dans le Tableau 16. Taverna propose une définition complète des workflows ou des fragments de workflows sur le site Web MyExperiment[56]. CASAS capitalise le retour d'expérience des compositions, par l'intermédiaire d'une pondération par l'intermédiaire du critère de QoS affecté automatiquement aux services, sans aucune sollicitation de l'utilisateur.

Tableau 16: Réutilisation de la connaissance dans CASAS

	Réutilisation de la connaissance		
	Eléments réutilisés	**Technique de réutilisation**	**Automatisation**
Taverna	Composants, exemples, et fragments par encapsulation dans des composants	Recherche de mots-clés, copié/collé, dépôt de wworkflows et forum.	Non prévue
CASAS	Fragments, retour utilisateur	Coroespondances avec un historique de compositions + sélection du poids des critères dans la composition	Embarquée dans l'algorithme de composition

Tableau 17: Caractéristiques des plateformes d'exécution

		Plateforme d'exécution	
	Outil de support	**Déploiement**	**Moteur d'exécution**
Taverna	Gestion de versions, manuels, tutoriels et FAQ	Sur place et distribué	Business Process engine
CASAS	Pas d'outil disponible	Sur place et distribué	Code natif

Les différences portant sur les outils de support à l'utilisation des deux logiciels et les caractéristiques des plateformes d'exécution sont détaillées dans le Tableau 17. Taverna hérite de «*Business Process Engine*» le multithreading natif, un monitoring du déroulement de l'exécution du workflow en temps réel, et d'autres fonctionnalités annexes que le code natif de CASAS ne propose pas dans sa version actuelle. Les outils de support à l'utilisation de CASAS n'existent pas, puisque le but de cette approche est précisément de permettre une utilisation la plus naturelle, la plus intuitive et la plus simple possible.

Un résumé général, présenté au Tableau 18 assigne à AstroTaverna et CASAS un jeu de critères communs plus globaux que les précédents. Le signe «++» désigne un critère qui fait partie des buts spécifiques à l'origine de la création de la plateforme, «+» indique que le critère est pris en compte par la plateforme et «-» identifie un critère pour lequel la plateforme ne propose aucune solution.

[56] [56] http://www.myexperiment.org

Tableau 18: Comparaison résumée AstroTaverna / CASAS

Critère	AstroTaverna	CASAS
Facilité d'utilisation générale	+	++
Facilité de découverte des services	+	++
Automatisation de la composition	-	++
Evaluation de la qualité durant la composition	-	++
Documentation du workflow	++	-
Réutilisation du workflow avec des paramètres différents	++	+
Contrôle de la composition par l'utilisateur	++	+

Comparativement à Taverna, CASAS propose une composition plus immédiate, et plus accessible. Les compositions de CASAS sont moins paramétrables puisqu'elles ne permettent pas de spécifier un service particulier à utiliser. Toutefois, la capitalisation du retour d'expériences sous forme de pondération des services permet d'améliorer la qualité des compositions proposées.
CASAS permet d'utiliser automatiquement tous les services contenus dans ASON, en fonction de l'adéquation de leurs capacités avec les besoins de la composition. CASAS apporte donc un gain d'automatisation, de simplicité d'utilisation et une garantie d'exhaustivité. Ces améliorations se font au prix de la documentation des workflows proposés, puisque CASAS n'indique que la description des services utilisés et les valeurs retournées par chacun de ces services, là où Taverna documente toute la chaîne de traitement. La raison en est que les workflows de CASAS ne sont pas destinés à être réutilisés en-dehors de l'application elle-même, mais destinés à améliorer les futures compositions. Il est donc inutile de ce point de vue de fournir une vue complète, qui permet dans Taverna d'identifier les points d'entrée de modifications éventuelles.

5.6. Conclusion

Suivant les principes du Web sémantique, CASAS déplace la responsabilité de la connaissance des capacités des services de l'utilisateur vers l'ontologie. L'application propose à la demande une composition sémantique automatique de services, sur la base d'une description des informations recherchées. Cette composition ne nécessite pas de se conformer à un langage de description particulier et ne demande pas d'intervention de l'utilisateur. La sélection des services adaptés parmi tous les services disponibles se fait en se basant sur les pondérations des services issues du retour d'expériences.

Etant donné que CASAS vise une automatisation complète de son fonctionnement, certaines interactions doivent être masquées. C'est notamment le cas lorsque des services d'analyse automatique ou de scripts sont invoqués ; les paramètres internes qui ne sont pas des sorties de services amont sont inaccessibles à l'utilisateur final et doivent être fixés au préalable. Une autre limitation vient du fait que pour utiliser un workflow composé par CASAS sur un ensemble donné d'objets astrophysiques, il est nécessaire de ré-exécuter l'invocation des services pour chaque objet de l'ensemble. D'autres gestionnaires de workflows (notamment Taverna) autorisent l'utilisateur à spécifier une liste de cibles et à interroger les services dans leur ensemble, sur l'intégralité de cette liste. L'application CASAS est

actuellement hébergé par une machine virtuelle avec un espace de stockage limité (20 Go) et un faible espace mémoire (RAM 4 Go) sur un processeur Intel (R) Xeon (R) CPU E5-2640 0 @ 2.50GHz. En conséquence, la composition et l'exécution de grands workflows sont lentes et peuvent durer plusieurs minutes, voire plusieurs dizaines de minutes dans certains cas d'exécution de « Deep workflows » importants.

Conclusion

1. Contributions

Nous avons présenté dans ce manuscrit une méthodologie permettant la description de services hétérogènes dans une ontologie de services commune, supportant une composition sémantique automatique de ces services. Cette composition automatique peut être utilisée pour répondre aux besoins d'utilisateurs sans connaissance technique ou sémantique des services composés, et assure une qualité de services basée sur les retours des utilisateurs et l'historique des compositions précédentes. La méthodologie proposée s'articule autour :

- **de la spécification et de la conception d'une ontologie globale pour la description de services et de leur domaine d'application**

Cette ontologie globale de services est organisée autour d'un module générique de description de services GEOS et d'un module thématique décrivant le contexte applicatif des services. Cette ontologie de services permet de fédérer des descriptions de services hétérogènes, et sert de support à une composition sémantique automatique de services. Nous avons pris le terme d'ontologie de services dans son sens le plus large, en y incluant non seulement des services Web mais plus globalement, tout appel à un programme informatique automatisé. Nous avons proposé dans le module générique GEOS une modélisation permettant d'exprimer des conditions complexes sur les entrées des services :

- Entrées caractérisées par une combinaison d'éléments portant sur une description, une ou plusieurs unités et un ou plusieurs formats
- Entrées multiples comprenant des sous-ensembles de combinaisons obligatoires et des sous-ensembles optionnels
- Entrées corrélées entre elles, impliquant des conditions sur la provenance des informations passées aux services

Le concept « QuerySoftware » introduit dans l'ontologie permet d'assurer l'automatisation réelle des appels aux services, indépendamment de leur profil technique (script, service Web, base de données…). Il fait pour cela référence à des briques logicielles générales (encapsulant tous les services partageant un protocole d'accès identique) ou spécialisées (encapsulant l'appel à des bibliothèques spécifiques).

- **d'une méthode pour obtenir une représentation quantifiée qualitativement d'un domaine d'application de services qui est peu couvert par les ontologies préexistantes**

Cette méthode permet d'obtenir un module thématique exprimant convenablement un contexte applicatif particulier, décrit dans des termes destinés à des experts du domaine et sans couverture ontologique préexistante importante, est également décrite dans ce manuscrit. Nous avons proposé une méthode d'extraction de connaissances structurées à partir d'une formalisation minimale de la

connaissance du domaine d'application et de textes courts, grammaticalement non structurés et à destination d'experts. Cette méthode repose sur l'apprentissage non supervisé (clustering), des mesures de similarité syntaxiques et l'analyse de la structure des textes par NLP. Une estimation de la confiance accordée à chacune des extractions faites de ce corpus est donnée, permettant de quantifier la validité estimée de ces concepts. Les deux modules de l'ontologie sont liés par la référence faite dans le module de description de services à des entrée et des sorties décrites dans le module thématique.

- **d'une méthode de composition et d'orchestration automatique de services basée sur la structure ontologique découlant des deux points précédents**

La composition sémantique automatique de services est la principale application de l'ontologie proposée. Il ne s'agit pas uniquement de composition sémantique de services Web mais de composition sémantique de services au sens général. Lorsque le domaine d'application fournit une infrastructure informatique distribuée existante et performante, cette composition de services doit s'adapter à cet existant et prendre en compte ses spécificités. Dans le cas où cette architecture propose des protocoles d'accès aux données indépendants de la sémantique des données proposées par les services, la composition ne peut pas se baser sur les seules exigences de ces protocoles. Il faut procéder par l'identification des informations fournies par les services, assurée par l'utilisation d'un algorithme adéquat. Dans le cas de services dont les contours sont bien définis mais dont le contenu est incertain, il est important de garder une trace des compositions réussies afin de réutiliser des sous-ensembles de ces compositions dans des contextes voisins. Il est également important de prendre en compte les retours d'expérience des utilisateurs experts du domaine, pour assurer une composition maximisant la qualité générale des services employés. La dernière exigence importante consiste à fournir la possibilité de matérialiser les résultats de la composition proposée par sa mise en oeuvre concrète et la production des résultats attendus. Cela ne peut se faire qu'à l'aide de la représentation des détails techniques des services utilisée par un algorithme d'orchestration et d'interrogation adapté. Les algorithmes capables de parcourir la représentation des connaissances proposée dans ce manuscrit y sont également décrits.

- **de l'application des concepts et méthodes proposés dans le cadre des services astrophysiques**

Les concepts et les méthodes de ce mémoire visent à amener des possibilités nouvelles, liées au Web sémantique dans l'architecture de l'OV. Viser cette forme d'interopérabilité émergente impose de se doter d'une architecture capable d'exprimer la sémantique propre au domaine d'application concerné. Nous proposons donc ASON comme une telle architecture, permettant l'intégration de services Web astrophysique au sein du Web sémantique. ASON est disponible en téléchargement[57] et sous-tend une application[58] visant à sélectionner, interroger et exécuter des outils et des services Web

[57] http://cta1.bagn.obs-mip.fr/ASONv1.0.owl

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

astrophysiques. Cette application appelée CASAS propose une découverte de service basée sur la description des résultats recherchés, en élargissant les possibilités existantes dans les logiciel compatibles OV et les interfaces d'annuaires disponibles. CASAS est un premier pas dans la direction de la composition automatique de services Web sémantiques appliquée à l'astrophysique. Nous proposons une composition entièrement automatisée des services basée sur une description sémantique de leurs capacités et de leur aspects techniques (*grounding*, URLs…). CASAS facilite la description des exigences portant sur le résultat des workflows, et vise à améliorer la sélection automatique des services par rapport à des méthodes basées sur les mots clés utilisées par les outils existants. La composition Web sémantique proposée par CASAS est une démarche qui mérite d'être testée, car elle apporte une nouvelle méthode pour définir automatiquement des compositions de services astrophysiques. La composition des workflows deviendra plus performante au fur et à mesure de son utilisation, grâce au TLL (*Traceability / Learning Layer*) qui mémorise les retours des utilisateurs et l'historique des combinaisons de services réussies.

2. Limites

2.1. Limites scientifiques

L'ontologie présentée dans ce manuscrit est une ontologie de services comportant un module thématique relatif au domaine de connaissances. Ce n'est pas une ontologie de domaine, et des questions portant sur la connaissance générale du domaine plutôt que sur les compétences précises des services ne peuvent pas être résolues par notre approche.

La définition automatique de la structure de l'ontologie n'est pas parfaite, des concepts identifiés avec des scores de précision faibles demeurent. D'autres concepts identifiés dans des contextes différents et mal réunifiés peuvent causer une séparation de l'information introduisant des erreurs dans l'identification de certains concepts. Ces problèmes d'identification peuvent aussi être la cause de rapprochements parcellaires entre les informations recherchée dans l'ontologie et son contenu réel. Purger l'ontologie de ces défauts passera par l'application de mécanismes d'alignement d'ontologies à l'intérieur de sa propre structure, pour identifier ces concepts redondants.

La composition de services repose sur l'identification d'individus qui sont les capacités des services répondant aux exigences du workflow demandé. Cette identification peut être longue, car elle balaye tous les individus de l'ontologie. L'algorithme de composition de services pourrait être amélioré en parcourant la structure hiérarchiquement à partir des branches dont la sémantique est la plus proche des exigences, pour aller vers les individus de branche en branche, plutôt que de feuille en feuille.

2.2. Limites applicatives

L'application fournie propose une sortie des informations résultant de l'exécution du workflow dans un navigateur Web. Pour pouvoir s'interfacer de façon efficace avec d'autres acteurs de l'OV, cette sortie devrait pouvoir se faire sous un format compatible OV (VOTable par exemple), correctement annoté avec les définitions OV les plus proches de la sémantique des résultats exprimés.

En astrophysique, un objet peut avoir plusieurs noms ou identifiants. CASAS s'appuie sur les services eux-mêmes pour la désambiguïsation des noms de cible que ses algorithmes ne fournissent pas. Ce problème de noms multiples pour une même cible est généralement résolu par les services eux-mêmes, comme les bases de données SOPHIE [59] ou Polarbase [60]. Les protocoles VO comme ConeSearch ou SSAP, utilisent quant à eux les coordonnées des objets au lieu de leurs identifiants.

Pour cette première version, les entrées des compositions de CASAS sont définies, et seules leurs valeurs sont libres. De plus, toujours dans cette première version CASAS propose d'indiquer un seul nom de cible en entrée de la composition.

Une autre limite vient de la gestion des unités dans lesquelles sont exprimées les quantités scientifiques retournées par les services. CASAS ne propose actuellement pas de conversion d'unités, et même si l'intégration de services de conversion ne diffère pas de l'intégration de tout autre service, elle reste à effectuer et à tester.

3. Perspectives

Les perspectives qu'offre le travail présenté dans ce manuscrit sont d'ordre scientifique et technique. Tout d'abord, il s'agit d'après nos connaissances du premier travail visant à amener une composition de services sémantique pour des services hétérogènes dans leur profil technique, et pas uniquement composée de services Web. Des perspectives d'élargissement de ce schéma peuvent être envisagées, même en restant dans le domaine d'application astrophysique. Il devient envisageable de coupler le pilotage d'instruments scientifiques dont le contrôle/commande est construit sur la base d'ontologies (Pessemier et al. 2015) aux travaux présentés dans ce manuscrit. Par exemple, lancer une observation sur le ciel en fonction des « trous » observationnels constatés en rapport avec les compétences de l'instrument peut être un cas d'utilisation.

Nous avons mentionné dans les limites, que l'ontologie proposée dans ce mémoire n'est pas une ontologie de domaine. Raisonner sur un problème scientifique général à partir d'une connaissance du domaine d'un haut niveau d'abstraction, amenant à une interrogation sur les capacités des services disponibles est une piste scientifique prometteuse. Cette piste implique la production d'une ou plusieurs ontologies décrivant le domaine ou les sous-domaines d'application. A terme, suivre cette direction pour le cas d'application présenté dans ce manuscrit pourrait amener à la création d'une structure comparable à un «WordNet pour l'astrophysique» propre à améliorer les performance, la fiabilité et les capacités non seulement d'ASON mais aussi des ontologies connexes à venir. Cette amélioration de la couverture ontologique du domaine astrophysique serait à même de soutenir des raisonnements susceptibles de poser les fondations d'un véritable système d'aide à la décision pour les

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

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

astrophysiciens. Un travail d'ingénieur suivi d'une thèse poursuivant les travaux présentés dans ce manuscrit est lancé, engageant une collaboration entre des chercheurs dans les domaines de l'informatique et de l'astrophysique.

ASON peut être réutilisée dans différents domaines scientifiques en dehors de l'astrophysique (géophysique, astrochimie ...) qui peuvent rencontrer les mêmes difficultés d'adaptation aux ontologies de services existantes et des problèmes semblables de descriptions de services. Des cas concrets d'adaptation du module thématique et de réutilisation du module de description de services méritent d'être étudiés.

D'un point de vue applicatif et technique, l'intégration d'une interface Web permettant à l'utilisateur final de fournir une description des nouveaux services dans CASAS est prévue. Cela amènera automatiquement de nouveaux services dans les descriptions d'ASON et permettra leur sélection pour des compositions futures. Le TLL est actuellement composé de fichiers, stockant chacun un certain nombre de paramètres d'historique. Matérialiser cette connaissance de l'historique dans l'ontologie elle-même ou dans une ontologie connexe est à l'étude.

L'amélioration de l'architecture technique est également prévue, et grâce à ces évolutions l'utilisateur gagnera la possibilité de spécifier plus d'entrées que ce qui est actuellement disponible. Il sera également possible de spécifier ses propres entrées avec son propre vocabulaire, comme pour les sorties. Préciser les unités souhaitées pour les sorties du workflow est également prévu, ainsi que la possibilité de passer une liste de noms de cibles plutôt qu'une seule cible.

Exprimer les résultats de la composition au format VOTable est une action planifiée, de même que la possibilité de retourner plus de sorties que les quatre sorties actuellement disponibles dans l'interface Web de CASAS.

Bibliographie

Abb, S. Ben et al., 2012. Characterizing Modular Ontologies D ' Aquin To cite this version :

Accomazzi, a et al., 2014. The Unified Astronomy Thesaurus. *ASP Conference Series*, (2009), pp.1–4.

Adala, A., Tabbane, N. & Tabbane, S., 2011. A framework for automatic web service discovery based on semantics and NLP techniques. *Advances in Multimedia*, 2011.

Akkiraju, R. et al., 2005. Web Service Semantics - WSDL-S. *W3C Workshop on Frameworks for Semantics in Web Services*, pp.1–42. Available at: http://www.knoesis.org/library/resource.php?id=00109.

Amudhavel, J. et al., 2016. Survey and Analysis of Web Service Composition Strategies: A State of Art Performance Study. *Indian Journal of Science and Technology*, 9(11). Available at: http://www.indjst.org/index.php/indjst/article/view/89265.

Antipolis, S., 2013. Bacem WALI Interopérabilité sé- mantique entre les outils de traitement d ' images en neu- roimagerie.

Arp, R. & Smith, B., 2011. Realizable Entities in Basic Formal Ontology. *International Classification*, (November), pp.1–6.

Ashburner, M. et al., 2000. Gene Ontology: Tool for The Unification of Biology. *Nature Genetics*, 25(1), pp.25–29.

Astrakhantsev, N. a. & Turdakov, D.Y., 2013. Automatic construction and enrichment of informal ontologies: A survey. *Programming and Computer Software*, 39(1), pp.34–42. Available at: http://link.springer.com/article/10.1134/S0361768813010039%5Cnhttp://dl.acm.org/citation.cfm?id=2434866.2434895%5Cnhttp://link.springer.com/10.1134/S0361768813010039.

Baccigalupo, C. & Plaza, E., 2007. Poolcasting: A social Web radio architecture for group customisation. *Proceedings - 3rd International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution, AXMEDIS 2007*, SS-97-06, pp.115–122. Available at: http://oa.upm.es/5484/.

Bansal, S. et al., 2014. Generalized semantic Web service composition. *Service Oriented Computing and Applications*, 10(2), pp.111–133.

Bartalos, P., Bieliková, M. & others, 2011. Automatic dynamic web service composition: A survey and problem formalization. *Computing and Informatics*, 30(4), pp.793–827.

Batista, D.C.T., 2011. SHOP2 : An HTN Planning System. *System*, 20, pp.3–6.

Bensaber, D.A. & Malki, M., 2012. Model driven approach for specifying WSMO ontology. In *CEUR Workshop Proceedings*. pp. 203–213.

Bentley, R. et al., 2013. HELIO: Discovery and analysis of data in heliophysics. *Future Generation Computer Systems*, 29(8), pp.2157–2168. Available at:

http://dx.doi.org/10.1016/j.future.2013.04.006.

Berneers-Lee, T., Hendler, J. & Lassila, O., 2001. The Semantic Web. *Scientific American*, 284(5), pp.34–43.

Bizer, C. et al., 2007. DBpedia: A Nucleus for a Web of Open Data. , pp.722–735. Available at: http://www.cis.upenn.edu/~zives/research/dbpedia.pdf.

Blomqvist, E., 2009. OntoCase-automatic ontology enrichment based on ontology design patterns. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5823 LNCS(2003), pp.65–80.

Boch, T. & Fernique, P., 2011. Aladin : An Open Source All-Sky Browser. , 442(2010), pp.683–686.

Carlson, A. et al., 2010. Coupled Semi-Supervised Learning for Information Extraction.

Chen, H. et al., 2015. Design of Automatic Extraction Algorithm of Knowledge Points for MOOCs. *Computational Intelligence and Neuroscience*, 2015.

Costa, A. et al., 2015. An Innovative Science Gateway for the Cherenkov Telescope Array. *Journal of Grid Computing*, 13(4), pp.547–559.

Cronin, K.A. et al., 2011. Behavioral response of a chimpanzee mother toward her dead infant. *American Journal of Primatology*, 73(5), pp.415–421. Available at: http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/download/1879/2201.

Debruyne, C. & Meersman, R., 2012. GOSPL: A method and tool for fact-oriented hybrid ontology engineering. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7503 LNCS, pp.153–166.

Derriere, S., Martinez, A.P. & Richard, A., 2010. Ontology of Astronomical Object Types. *Building*, pp.1–34.

Doran, P., 2005. Ontology reuse via ontology modularisation. *Computer*, pp.1–6.

Dowler, P. et al., 2014. IVOA Recommendation: DALI: Data Access Layer Interface Version 1.0. *CoRR*, abs/1402.4. Available at: http://arxiv.org/abs/1402.4750.

Dubernet, M.L. et al., 2016. The virtual atomic and molecular data centre (VAMDC) consortium. *J. Phys. B: At. Mol. Opt. Phys. Journal of Physics B: Atomic, Molecular and Optical Physics J. Phys. B: At. Mol. Opt. Phys*, 49(49), pp.74003–18. Available at: http://iopscience.iop.org/0953-4075/49/7/074003%5Cnhttp://www.vamdc.eu.

Dwyer, S.F., 2013. An Analysis of Ontology Engineering Methodologies: A Literature Review. , 6(16), p.8 TS-CrossRef.

Essaid, A. et al., 2012. Gestion du conflit dans l ' appariement des ontologies To cite this version : , pp.50–60.

Evans, R., 2003. A Framework for Named Entity Recognition in the Open Domain. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, 260(267–274), p.267. Available at: http://books.google.co.kr/books?id=ejXv2scn794C%5Cnhttp://clg.wlv.ac.uk/papers/evans-RANLP-03.pdf.

Faria, C., Girardi, R. & Novais, P., 2013. Analysing the problem and main approaches for ontology population. *Proceedings of the 2013 10th International Conference on Information Technology: New Generations, ITNG 2013*, (1), pp.613–618.

Farrell, J. & Lausen, H., 2007. Semantic Annotations for WSDL and XML Schema (W3C Recommendation 2007). *Recommendation, W3C*, (August). Available at: https://www.w3.org/TR/sawsdl/.

Fernández-López, M., Gómez-Pérez, A. & Juristo, N., 1997. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. *AAAI-97 Spring Symposium Series*, SS-97-06, pp.33–40. Available at: http://oa.upm.es/5484/.

Fielding, R.T., 2000. Architectural Styles and the Design of Network-based Software Architectures. *Building*, 54, p.162. Available at: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.

Frey, B.J. & Dueck, D., 2007. Clustering by passing messages between data points. *Science (New York, N.Y.)*, 315(5814), pp.972–976. Available at: http://www.ncbi.nlm.nih.gov/pubmed/17218491.

Gangemi, A. et al., 2005. Ontology evaluation and validation. *Media*, 3, pp.1–53.

Gruber, T.R., 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), pp.199–220. Available at: http://www.sciencedirect.com/science/article/pii/S1042814383710083.

Haase, P. et al., 2008. The neon ontology engineering toolkit. *Www*, (April), pp.4–6. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:The+NeOn+Ontology+Engineering+Toolkit#0.

Halevy, A. & Noy, N., 2016. Discovering Structure in the Universe of Attribute Names. *Www*, pp.939–949.

Hlomani, H. et al., 2011. Utilizing a compositional system knowledge framework for ontology evaluation: A case study on BioSTORM. *KEOD 2011 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, pp.167–175. Available at: http://www.scopus.com/inward/record.url?eid=2-s2.0-84862160111&partnerID=tZOtx3y1.

Hlomani, H. & Stacey, D., 2014. Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semantic Web Journal*, 1, pp.1–11. Available at: http://www.semantic-web-journal.net/system/files/swj657.pdf.

Hucka, M. et al., 2015. Promoting Coordinated Development of Community-Based Information Standards for Modeling in Biology: The COMBINE Initiative. *Frontiers in bioengineering and biotechnology*, 3(February), p.19. Available at: http://journal.frontiersin.org/article/10.3389/fbioe.2015.00019/abstract.

Kamaruddin, L.A., Shen, J. & Beydoun, G., 2012. Evaluating usage of WSMO and OWL-S in semantic web services. *Conferences in Research and Practice in Information Technology Series*, 130, pp.53–58.

Karger, D. et al., 2004. The Semantic Web – ISWC 2004. *The Semantic Web–ISWC 2004*, 3298, pp.214–228. Available at: http://www.springerlink.com/content/lcqgewykavhvle8y/.

Karray, M.H. et al., 2012. A Formal Ontology for Industrial Maintenance . To cite this version : A Formal Ontology for Industrial Maintenance.

Lantow, B. & Sandkuhl, K., 2015. An Analysis of Applicability using Quality Metrics for Ontologies on Ontology Design Patterns. *Intelligent Systems in Accounting, Finance and Management*, 22(1), pp.81–99.

Lèbre, A. et al., 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*. pp. 365–368.

Lee, T. et al., 2013. Attribute extraction and scoring: A probabilistic approach. *Proceedings - International Conference on Data Engineering*, pp.194–205.

Lemos, A.L., Daniel, F. & Benatallah, B., 2016. Web service composition: a survey of techniques and tools. *ACM Computing Surveys (CSUR)*, 48(3), p.33.

Levenshtein, V.I., 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8), pp.707–710.

Li, Y. et al., 2006. Sentence Similarity Based on Semantic Nets and Corpus Statistics. , 18(8), pp.1–35.

López, F., 1999. Overview Of Methodologies For Building Ontologies. *Proceedings of the IJCAI99 Workshop on Ontologies and ProblemSolving Methods Lessons Learned and Future Trends CEUR Publications*, 1999(2), pp.1–13. Available at: http://iwayan.info/Research/Ontology/Tutor_Workshop/Tutorial_4_Analysis.pdf.

Makarov, D. et al., 2014. HyperLEDA. III. The catalogue of extragalactic distances. *Astronomy & Astrophysics*, 570, p.A13. Available at: http://adsabs.harvard.edu/abs/2014A%26A...570A..13M.

Martin, D. et al., 2007. Bringing semantics to web services with OWL-S. *World Wide Web*, 10(3Martin, D. et 2007. Bringing semantics to web services with OWL-S. World Wide Web, 10(3), p.243–277.), pp.243–277.

Mausam et al., 2012. Open Language Learning for Information Extraction. *EMNLP-CoNLL*, (July), pp.523–534.

McIlraith, S., Son, T. & Zeng, H., 2001. Semantic Web Services. *IEEE Intelligent Systems. Special Issue on the Semantic Web*, 16(2), pp.46 – 53.

Mikolov, T. et al., 2013. Efficient Estimation of Word Representations in Vector Space. , pp.1–12. Available at: http://arxiv.org/abs/1301.3781.

Milanovic, N. & Malek, M., 2004. Current solutions for Web service composition. *IEEE Internet Computing*, 8(6), pp.51–59. Available at: http://link.springer.com/10.1007/978-1-4614-6170-8_132.

Miller, G. a., 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11), pp.39–41.

Nicola, B.Y.A.D.E., Missikoff, M. & Living, W.E.A.R.E., 2016. Methodology for Rapid Ontology Engineering. *Communications of the ACM*, pp.79–86.

Nieto, F.J., 2011. Enterprise Interoperability. *Lecture Notes in Business Information Processing*, 76(MARCH), pp.118–131. Available at: http://www.scopus.com/inward/record.url?eid=2-s2.0-79953058456&partnerID=tZOtx3y1.

O'Connor, M. & Das, A., 2009. SQWRL: A query language for OWL. *CEUR Workshop Proceedings*, 529(January 2009).

Parsia, B., 2003. Semantic Web Services. *Bulletin of the American Society for Information Science*, 16(May), pp.12–15.

Pedrinaci, C. et al., 2010. IServe: A linked services publishing platform. *CEUR Workshop Proceedings*, 596, pp.71–82.

Pedrinaci, C. & Domingue, J., 2010. Toward the Next Wave of Services: Linked Services for the Web of Data. *Journal Of Universal Computer Science*, 16(13), pp.1694–1719. Available at: http://people.kmi.open.ac.uk/carlos/wp-content/uploads/downloads/2010/09/services-for-the-web-of-data.pdf.

Pessemier, W. et al., 2015. Why Semantics Matter : a Demonstration on Knowledge-Based Control System Design. , pp.9–12.

Pia, M., 2015. Information Extraction for Ontology Population Tasks . An Application to the Italian Archaeological Domain. , 3(2), pp.40–50.

Porzel, R. & Malaka, R., 2004. A Task-based Approach for Ontology Evaluation. In *ECAI Workshop on Ontology Learning and Population*.

Poveda-villalón, M., Suárez-figueroa, M.C. & Gómez-pérez, A., 2012. Validating Ontologies with OOPS ! State of the Art. , pp.267–268.

Presutti, V. & Gangemi, A., 2008. Content Ontology Design Patterns as Practical Building Blocks forWeb Ontologies. *Conceptual Modelling - ER 2008*, 5231, pp.128–141. Available at: http://dl.acm.org/citation.cfm?id=1478324.1478339.

Puttonen, J., Lobov, A. & Lastra, J.L.M., 2013. Semantics-based composition of factory automation processes encapsulated by web services. *IEEE Transactions on Industrial Informatics*, 9(4), pp.2349–2359.

Richard, M. et al., 2015. ethode interactive pour la validation d ' ontologies To cite this version : LOVMI : vers une méthode interactive pour la validation d ' ontologies.

Rodriguez-Mier, P. et al., 2016. An integrated semantic web service discovery and composition framework. *IEEE Transactions on Services Computing*, 9(4), pp.537–550.

Roman, D. et al., 2015. WSMO-Lite and hRESTS: Lightweight semantic annotations for Web services and RESTful APIs. *Journal of Web Semantics*, 31, pp.39–58. Available at: http://dx.doi.org/10.1016/j.websem.2014.11.006.

Ruiz, J.E. et al., 2014. {AstroTaverna}: Building workflows with {Virtual Observatory} services. *Astronomy and Computing*, 7--8, pp.3–11. Available at: http://www.sciencedirect.com/science/article/pii/S2213133714000419.

Sarkissian, A. et al., 2016. The International Planetary Data Alliance (IPDA): Overview of the Activities. , 505, pp.29–34.

Sbodio, M.L., Martin, D. & Moulin, C., 2010. Discovering Semantic Web Services using SPARQL and Intelligent Agents. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4), pp.310–328. Available at: http://www.sciencedirect.com/science/article/pii/S1570826810000533.

Severi, P., Fiadeiro, J. & Ekserdjian, D., 2010. Guiding reification in OWL through aggregation. *CEUR Workshop Proceedings*, 573(January 2010), pp.408–419.

Shafiq, O., 2007. Triple space computing for Semantic Web services. In *CEUR Workshop Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 83–84. Available at: http://dx.doi.org/10.1007/978-3-642-19193-0_7.

Shvaiko, P. & Euzenat, J., 2013. Ontology matching: State of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1), pp.158–176. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6104044.

Silberztein, M., 1993. *Dictionnaires électroniques et analyse automatique de textes: le système INTEX*, Masson. Available at: https://books.google.tn/books?id=Q2SAPQAACAAJ.

Simon, L. et al., 2005. A Universal Service Description Language. *Web Services, IEEE International Conference on*, 0, pp.823–824. Available at: http://www.computer.org/plugins/dl/pdf/proceedings/icws/2005/2409/00/24090823.pdf?template=1&loginState=2&userData=Friedrich-Althoff-Consortia+Berlin-Brandenburg+%2528consortium%2529+%257E+Universitaet+Leipzig%253AFriedrich-Althoff-Consortia+Berlin-Brande.

Smith, B. et al., 2007. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25(11), pp.1251–1255.

Sofiyanti, N., Fitmawati, D.I. & Roza, A.A., 2015. Stenochlaena Riauensis (Blechnaceae), A new fern species from riau, Indonesia. *Bangladesh Journal of Plant Taxonomy*, 22(2), pp.137–141. Available at: http://www.jmlr.org/papers/v12/pedregosa11a.html.

Suchanek, F.M., Kasneci, G. & Weikum, G., 2007. Yago. *Proceedings of the 16th international conference on World Wide Web - WWW '07*, p.697. Available at: http://portal.acm.org/citation.cfm?doid=1242572.1242667.

Tartir, S. et al., 2005. OntoQA: Metric-Based Ontology Quality Analysis. *IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, pp.45–53.

Taylor, M.M. ~B., 2005. TOPCAT: Tool for OPerations on Catalogues And Tables. *Astronomical Data Analysis Software and Systems XIV*, 347(S 01010), p.29.

Thomas, B., 2015. Development of a VO Registry Subject Ontology using Automated Methods. , pp.1–4. Available at: http://arxiv.org/abs/1502.05974.

Tosi, D. & Morasca, S., 2015. Supporting the semi-automatic semantic annotation of web services: A systematic literature review. *Information and Software Technology*, 61, pp.16–32.

Tushkanova, O. & Gorodetsky, V., 2015. Data-driven semantic concept analysis for automatic actionable ontology design. *Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015*.

Uschold, M. et al., 1996. Ontologies : Principles , Methods and Applications. *Knowledge Engineering Review*, 11(2), pp.93–136.

Wache, H. et al., 2001. Ontology-Based Information Integration: A Survey of Existing Approaches. *International Joint Conference on Artificial Intelligence; Workshop: Ontologies and Information Sharing*, pp.108–117. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.7857.

Winkler, W.E., 1999. The State of Record Linkage and Current Research Problems. , p.15.

Xing, W. & Wenpeng, Z., 2010. An automatic Semantic Web Service composition approach based on PDDL. In *Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference on*. pp. 71–74. Available at: http://ieeexplore.ieee.org.ez67.periodicos.capes.gov.br/xpls/abs_all.jsp?arnumber=5532129.

Zhao, H., Lu, Z. & Poupart, P., 2015. Self-adaptive hierarchical sentence model. *IJCAI International Joint Conference on Artificial Intelligence*, 2015–Janua, pp.4069–4076. Available at: http://arxiv.org/abs/1301.3781.

Zhao, X. et al., 2013. Finding preferred skyline solutions for SLA-constrained service composition. In *Proceedings - IEEE 20th International Conference on Web Services, ICWS 2013*. pp. 195–202.

Zouaq, A. & Nkambou, R., 2008. Building domain ontologies from text for educational purposes. *IEEE Transactions on Learning Technologies*, 1(1), pp.49–62.