

Licence Informatique

La Rochelle Université

*Création de plugins gérant les appartenances floues
pour la plateforme GALACTIC*



Éléonore Thonneau

2022

Utilisation du portrait d'Évariste Galois réalisé par M. Yann Gautreau
aimablement autorisée dans un cadre universitaire

Table des matières

1	Résumé	2
2	Abstract	2
3	L'entreprise	4
4	La plateforme GALACTIC	6
5	Le stage	9
5.1	Le sujet de stage	9
5.2	La logique floue	9
6	Le déroulement du stage	11
6.1	Présentation et installation de la plateforme	11
6.2	Mise en place du plugin de caractéristiques	11
6.3	Mise en place du plugin de description	15
6.4	Mise en place du plugin de stratégie	16
6.5	Le travail réalisé	17
6.6	Le travail effectué	17
7	Les outils utilisés durant mon Stage	17
8	Conclusion	19
9	Annexe - Sources	20

Liste des figures

1	Logo du Laboratoire Informatique, Images et Interaction	4
2	Icône de la plateforme GALACTIC	6
3	Architecture de la plateforme GALACTIC.	8
4	Exemple d'un cas de la logique booléenne classique et d'un cas de la logique floue.	9
5	Exemple d'une fonction d'appartenance.	10
6	Logo représentant les données liées à logique floues, sur la plateforme GALACTIC.	10
7	Logo représentant les plugins de caractéristiques de la plateforme GALACTIC.	11
8	Logo représentant les plugins de description de la plateforme GALACTIC.	15
9	Logo représentant les plugins de stratégie de la plateforme GALACTIC.	16
10	Exemple de stratégie	17

1 Résumé

Je m'appelle Eléonore THONNEAU, étudiante en 3ème année de Licence Informatique, je suis en stage au sein du Laboratoire Informatique, Images et Interaction (L3i) de l'Université de La Rochelle.

Durant mon stage au L3i j'ai travaillé au sein de l'équipe Modèles et Connaissances, en particulier sur le projet de la plateforme GALACTIC. Cette dernière a pour objectif de mettre en place un framework permettant de travailler sur l'Analyse Formelle des Concepts, plus précisément sur l'algorithme NextPriorityConcept, ainsi que sur la Théorie des treillis.

Mon rôle sur ce projet était de mettre en place trois nouveaux plugins pour gérer les données liées à la logique floue :

- un plugin de caractéristiques
- un plugin de description
- un plugin de stratégie

2 Abstract

My name is Eléonore THONNEAU, a 3rd year Bachelor of Computer Science student, I am an intern in the "Laboratoire Informatique, Images et Interaction" (L3i) of the University of La Rochelle.

During my internship at L3i I worked on the "Modèles et Connaissances" team, especially on the GALACTIC platform project. The purpose of the latter is to set up a framework for working on the Formal Concept Analysis, specifically on the NextPriorityConcept algorithm, as well as on the Lattice Theory.

My role on this project was to implement three new plugins to manage data related to fuzzy logic:

- a features plugin
 - a description plugin
 - a strategy plugin
-

Remerciements

Je souhaite remercier toutes les personnes m'ayant aidé tout au long de mon stage.

Je tiens tout d'abord à remercier mon maître de stage Monsieur DEMKO pour le temps qu'il m'a accordé ainsi que toutes les explications et l'aide qu'il m'a apporté tout au long de mon stage afin de faciliter la compréhension du sujet et de la Plateforme sur laquelle j'ai travaillé durant mon stage, mais je le remercie également pour m'avoir aidé pour la rédaction de mon rapport de Stage.

Je remercie ma tutrice de stage Madame BERTET, pour avoir accepté ma demande de tutorat et pour les explications données vis-à-vis du déroulement d'un travail de recherche dans un Laboratoire, ainsi que pour les informations apportées pour la présentation de l'entreprise mais également pour la présentation de la plateforme GALACTIC, entre autre pour les explications de l'algorithme NextPriorityConcept.

Je remercie également toute l'équipe de la plateforme GALACTIC pour leur accueil au cours des différentes réunions d'équipe qui ont lieu au cours de mon stage au sein de L3i.

Je tiens à remercier Monsieur Yacine GHAMRI-DOUDANE, Responsable du L3i et Monsieur Muhammad Muzzamil LUQMAN, Responsable du recensement des stagiaires au sein du L3i, pour l'accueil chaleureux qu'ils m'ont témoigné au sein du Laboratoire Informatique, Image et Interaction durant toute la durée de mon stage.

Pour finir je remercie, Jules BRICOU, Alexandre BUFFARD, Kévin VOISIN et Martin EHLINGER, pour le soutien moral et l'aide qu'ils m'ont apporté pendant toute la durée de mon stage. Ainsi que pour notre rédaction commune de la présentation de l'entreprise et de la plateforme GALACTIC sur laquelle nous avons travaillé tous les cinq.

3 L'entreprise



Figure 1: Logo du Laboratoire Informatique, Images et Interaction

La présentation de l'entreprise est issue du site de l'entreprise (<https://l3i.univ-larochelle.fr/Présentation-317>) et il s'agit d'un travail fait en commun avec Jules BRICOU, Kévin VOISIN, Alexandre BUFFARD, Martin EHLINGER et moi-même.

Le Laboratoire Informatique, Image et Interaction (L3i) a été créé en 1993, près de 100 personnes travaillent au sein du laboratoire. Il s'agit d'un laboratoire de recherche de Sciences du Numérique de l'Université de La Rochelle, il associe les chercheurs en Informatique de l'IUT ainsi que les chercheurs du Pôle Sciences de l'Université de La Rochelle. Il compte 33 chercheurs et chercheuses, 35 doctorants et doctorantes, 4 personnes permanentes de soutien à la recherche et 25 personnes sur projets et est dirigé par Monsieur Yacine GHAMRI-DOUDANE.

Le laboratoire L3i fait parti de réseaux de recherches régionaux (Fédération CNRS MIRES, ERT " Interactivité Numérique "), nationaux (GDR I3 et GDR IRIS) et internationaux (IAPR) dans les secteurs de visibilité de son action scientifique, autour d'un projet stratégique lié à la gestion intelligente et interactive des contenus numériques. Cela est renforcé grâce à une politique de volontariste de participation ou de pilotages de projets de recherches labélisés (ANR, PCRD, ...), au sein desquels le laboratoire occupe couramment une position de leadership.

Le Laboratoire est également membre de l'Alliance Big data lancée en 2013 pour favoriser le développement, en France, de nouveaux services et projets dans ce domaine.

Ses travaux sont menés en partenariat avec une dizaine de centres de recherches nationaux (dont l'INRIA, institut spécialisé). Le L3i entretient par ailleurs des liens privilégiés avec de nombreux centres de recherche à travers le monde. Il est également engagé dans près d'une vingtaine de partenariats industriels sur l'ensemble du territoire français.

Le laboratoire du L3i est divisé en 3 équipes:

- Modèles et Connaissances
- Images et Contenus
- Dynamique des systèmes et Adaptativité

Mon stage se déroule au sein de l'équipe Modèles et Connaissances et plus précisément autour du projet GALACTIC de cette équipe.

4 La plateforme GALACTIC

La présentation de la plateforme GALACTIC est issue du diaporama de présentation de l'architecture de la plateforme (<https://galactic.univ-lr.fr/slides/architecture/Galactic-Architecture-slides-20min-complete.pdf>) et il s'agit d'un travail fait en commun avec Jules BRICOU, Kévin VOISIN, Alexandre BUFFARD, Martin EHLINGER et moi-même.

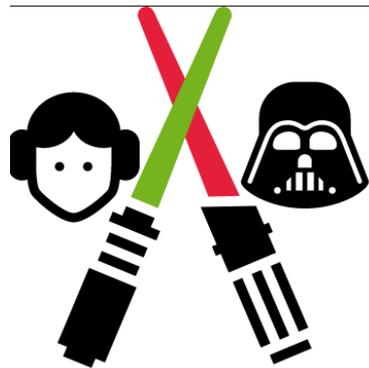


Figure 2: Icône de la plateforme GALACTIC

Le logo de la plateforme a été pensé de la façon suivante :

- Les deux sabres lasers croisés sont en fait un hommage au mathématicien Évariste Galois qui est mort au cours d'un duel galant lorsqu'il était âgé de vingt ans.
- Quant aux images représentant la Princesse Leia Organa et Dark Vader de Star Wars, c'est pour représenter le terme "GALACTIC".

GALACTIC est l'acronyme de **G**alois **L**attices **C**oncept **T**heory **I**mplicational system and **C**losures. L'objectif de la plateforme est pouvoir mettre en place un framework permettant de travailler sur l'Analyse Formelle des Concepts ainsi que la Théorie des treillis.

L'Analyse Formelle des Concepts implique la définition de concepts (paires formées par un objet et un attribut) dans un contexte donné, ils sont tous deux complètement et précisément définis. L'Analyse Formelle des Concepts est utilisée pour la classification et le regroupement car elle permet l'association de plusieurs objets à une même description.

En mathématiques, les treillis sont des structures algébriques utilisés en algèbres. Il s'agit d'ensembles partiellement ordonnés dans lesquels chaque paire d'éléments admet une borne inférieure et une borne supérieure.

La plateforme GALACTIC est structurée de la façon suivante:

- Un noyau, représenté par le coeur de la fleur, qui contient les opérations de base et structures de données et qui implémente un nouvel algorithme NEXTPRIORITYCONCEPT inspiré des pattern structure.

- Des plugins, représentés par les pétales de la fleur:
 - Caractéristiques : Booléenne, Numérique, Catégorique, String, Séquentielle, Chain, Triadic. Ils permettent de définir les caractéristiques.
 - Descriptions : Booléenne, Logique, Catégorique, String (regex), String (distance), Chain, Séquentielle, Séquentielle (distance), Triadic. Ils définissent les prédicats et les espaces de description utilisés pour représenter et définir les données avec précision.
 - Stratégies : Booléenne, Logique, Catégorique, Numérique, String, String (distance), Chain, Séquentielle, Séquentielle (distance), Triadic. Ils définissent la manière utilisée pour explorer les données, il utilise des descriptions pour générer des prédécesseurs pour chaque concept dans le treillis.
 - Mesures : predecessor Cardinality, sucesor Cardinality, Confidence. Ce sont des paramètres des stratégies de filtrage prédéfinis dans la librairie core.
 - Data Readers : Ils sont utilisés pour lire différents types de fichiers de données. Le moteur de base détecte le type de fichier en utilisant son extension. Les plugins Data Readers existants sont: YAML, JSON, CSV, TOML, INI, TXT, SLF, DAT, CXT.
 - Localization: Ils sont utilisés pour traduire les applications dans d'autres langues.

- Des applications, représentées par les feuilles de la fleur, qui sont développées afin d'utiliser la librairie de la plateforme. Ce sont des interfaces pour l'utilisateur:
 - GALACTIC Laser : pour construire les treillis et explorer les données.
 - GALACTIC Explorer : pour explorer de façon interactive les treillis construits.
 - GALACTIC Ruler : pour extraire les règles d'implication.
 - GALACTIC Fire : pour exécuter un système de règles.

- La bulle au dessus du serpent indique les différentes versions de python avec lesquelles la plateforme GALACTIC est compatible.

- Au niveau des racines on peut voir les différentes langues disponibles sur la plateforme.

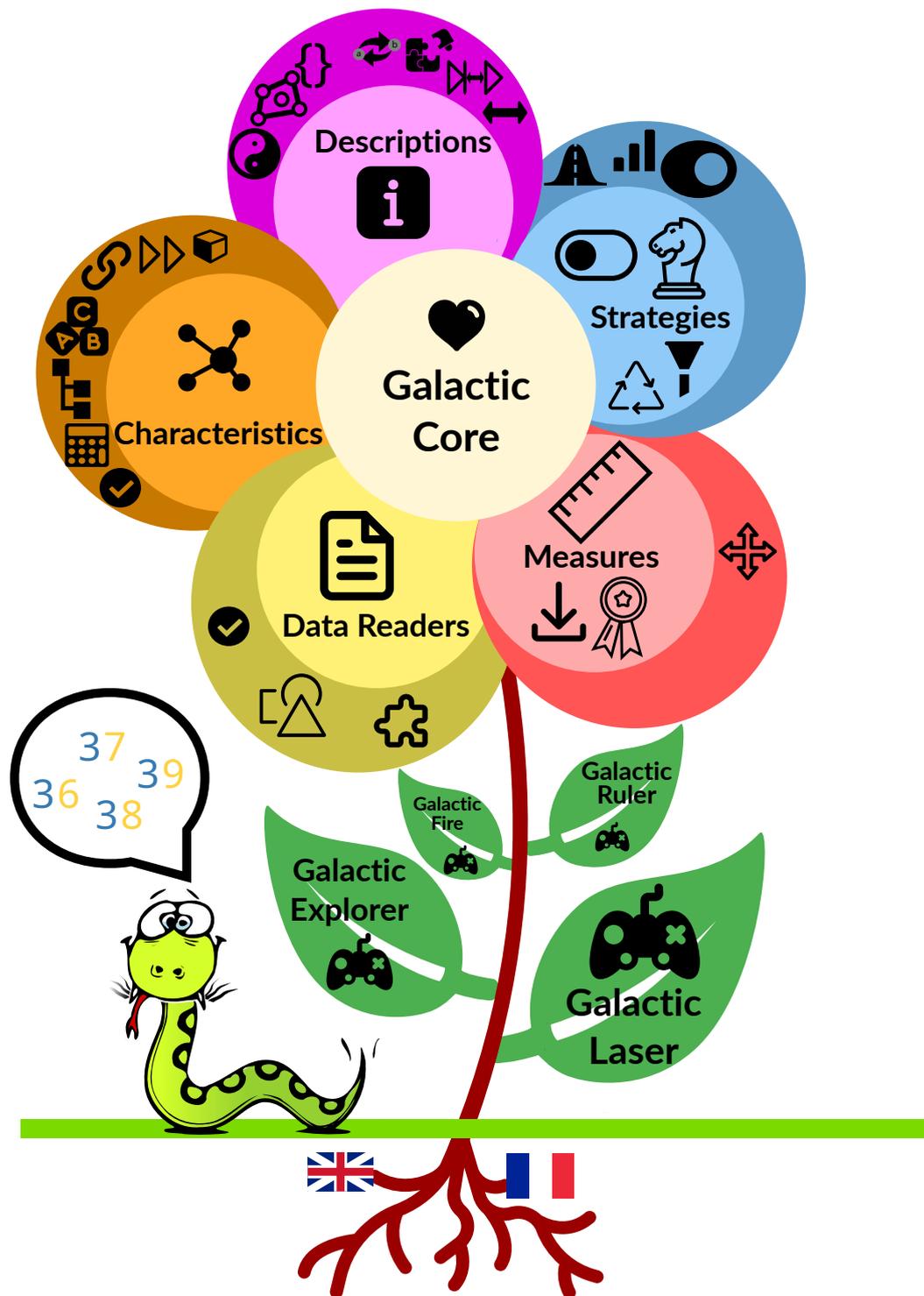


Figure 3: Architecture de la plateforme GALACTIC.

5 Le stage

5.1 Le sujet de stage

Mon stage s'est déroulé au sein du Laboratoire Informatique, Images et Interaction (L3i) de l'Université de La Rochelle, au sein de l'équipe Modèles et Connaissances. J'ai travaillé sur le projet de la plateforme GALACTIC présentée précédemment. Plusieurs plugins ont déjà été créés pour la plateforme GALACTIC afin de pouvoir travailler sur différents types de données telles que les données numériques, les données logiques, les données séquentielles, les données catégoriques ou encore les données de type *String*.

Il m'a donc été demandé de mettre en place trois nouveaux plugins pour la plateforme GALACTIC: un plugin de caractéristiques, un plugin de description et un plugin de stratégie permettant de gérer les données liées à la logique floue.

Tout d'abord un plugin de caractéristiques nous permet d'extraire les données souhaitées, ensuite un plugin de description nous permet de décrire les données que nous avons extraites, il s'agit d'un moyen simple de décomposer les données, et enfin un plugin de stratégie qui nous permet de séparer les données en plusieurs sous-groupes.

5.2 La logique floue

La logique floue, liée à la théorie des ensembles flous ainsi qu'à la théorie des possibilités, est très similaire à la logique booléenne classique, mais contrairement à cette dernière qui n'admet que 0 ou 1 comme valeur, la logique floue admet tous les nombres réels qui se situent entre 0 et 1, ainsi nous pouvons considérer que la logique floue est une extension de la logique booléenne classique.

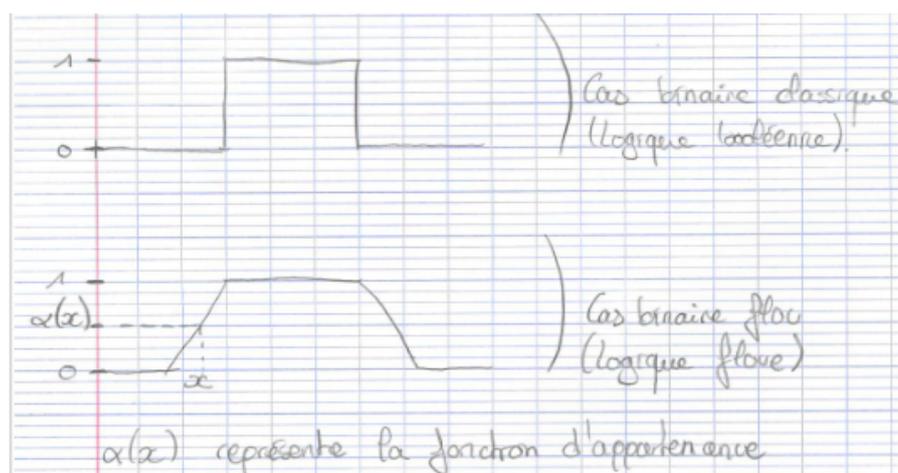


Figure 4: Exemple d'un cas de la logique booléenne classique et d'un cas de la logique floue.

Les éléments de base de la logique floue sont : les variables linguistiques, les fonctions d'appartenance et les déductions aux inférences.

- **Variables linguistique:** Une description d'une situation, d'un événement ou d'un procédé contient généralement des qualificatifs flous tels que: peu , beaucoup, énormément, froid, tiède, chaud, petit, moyen, grand, etc.
- **Fonctions d'appartenance:** Contrairement à la logique booléenne classique, où un ensemble donné appartient à l'ensemble "vrai" ou à l'ensemble "faux", la logique floue admet un degré d'appartenance, ce dernier étant représenté par un nombre compris entre 0 et 1. La valeur précise de la fonction d'appartenance liée à une valeur de la variable est notée μ et appelée "facteur d'appartenance".

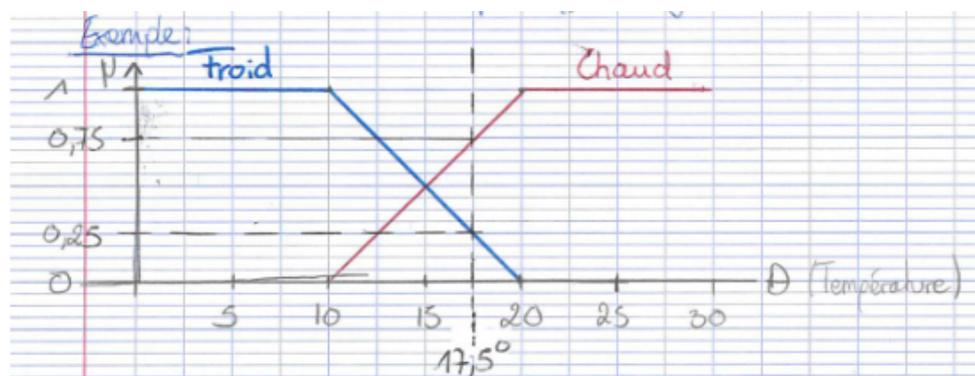


Figure 5: Exemple d'une fonction d'appartenance.

D'après l'exemple ci-dessus, pour une valeur $x = 17,5^\circ$ le facteur d'appartenance à l'ensemble froid vaut $\mu(\text{froid}) = 0,25$ et le facteur d'appartenance à l'ensemble chaud vaut $\mu(\text{chaud}) = 0,75$. Les fonctions d'appartenance peuvent avoir différentes formes, cependant elles sont généralement définies par des segments de droites.

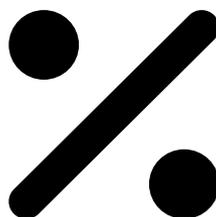


Figure 6: Logo représentant les données liées à la logique floue, sur la plateforme GALACTIC.

6 Le déroulement du stage

6.1 Présentation et installation de la plateforme

Le premier jour du stage Martin, Kévin et moi-même avons eu une présentation de l'architecture de la plateforme GALACTIC ainsi qu'une présentation de nos sujets de stage respectifs de la part de Monsieur DEMKO, il nous a par ailleurs donné des indices pour nous mettre sur la voie pour voir par où nous devons commencer. Le lendemain nous avons été conviés à une réunion avec l'équipe du projet GALACTIC, où nous avons eu une présentation de l'équipe ainsi qu'une présentation de l'algorithme NEXTPRIORITYCONCEPT présenté par Madame Bertet. Nous avons respectivement présenté nos sujets de stage à l'équipe, qui nous a aidé à mieux comprendre nos sujets.

Pour pouvoir commencer à travailler sur la création des nouveaux plugins pour la plateforme, il fallait faire l'installation de la plateforme GALACTIC sur mon ordinateur. Ayant un ordinateur sous Windows, j'ai suivi le guide d'installation pour Windows qui est disponible sur le site de la plateforme (<https://galactic.univ-lr.fr/docs/main/latest/install/index.html>), néanmoins malgré le suivi des différentes instructions inscrites dans le guide d'installation je n'ai malheureusement pas réussi à installer la plateforme sous Windows. Donc j'ai mis en place un dual boot Linux sur ma machine, avec l'aide de Martin et de Kévin craignant de commettre une erreur irréparable, afin de pouvoir installer la plateforme et utiliser les plugins et les applications déjà opérationnels de la plateforme. J'ai mis en place un dual boot Linux car Monsieur Demko nous avait dit qu'il n'y avait aucun problème d'installation avec le système d'exploitation Linux.

6.2 Mise en place du plugin de caractéristiques

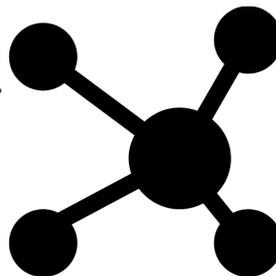


Figure 7: Logo représentant les plugins de caractéristiques de la plateforme GALACTIC.

Un plugin de caractéristiques nous permet d'extraire les données, ici nous souhaitons extraire des données liées à la logique floue.

Pour créer le plugin de caractéristiques permettant de gérer les données liées à la logique floue, j'ai tout d'abord effectué des recherches sur la logique floue afin de bien cerner le sujet et de pouvoir procéder au mieux pour la création du plugin. Afin de mettre en place ce dernier, j'ai utilisé une partie du code établi pour le plugin de caractéristiques des données numériques, étant donné que les valeurs

des données liées à la logique floue sont des nombres réels qui se situent entre 0 et 1 et donc sont des valeurs numériques. Voici la partie de code du plugin de caractéristiques numériques dont je me suis servie:



```
class Numerical(Imprecise):
    """ This class is only used to factorize some common code to all
    ↪ sub-classes """
    __module__ = "galactic_characteristic_numerical"
    """ It prints the name of the module in which the function or object
    ↪ was defined, or None if it is not available. """
    __slots__ = ()
    """ This attribute allows us to specify the instance attributes we
    ↪ expect from our object instance. Furthermore this allows a faster
    ↪ access to attribute and a saving space in memory. """
    def __call__(self, individual: Any) -> Optional[Collection[float]]:
        """ It is an abstract method which make this characteristic
        ↪ callable. It takes, in arguments, any python object and
        ↪ returns a collection of possible values. """
        raise NotImplementedError
        """ It indicates that the implementation still need to be added.
        ↪ """
```

J'ai conservé la classe tel qu'elle est, je n'ai fait que changer le nom de la classe par *Fuzzy* et rajouter des commentaires au niveau des attributs et des fonctions afin de me rappeler à quoi chacun correspondait et ce qu'ils faisaient.



```
class Number(MultipleValue, Numerical):
    """
    The :class:`Number` class is used to retrieve a number from an
    ↪ individual.
    Example
    -----
    >>> from galactic.characteristics import Key
    >>> from galactic_characteristic_numerical import Number
    >>> a = Number(characteristic=Key(name="x"))
    >>> print(a)
    x
    >>> a({"x": 5})
    [5.0]
    >>> a(4)
    """
    __module__ = "galactic_characteristic_numerical"
    __slots__ = ()
    yaml_tag = "!characteristic.numerical.Number"
```

```

def __init__(self, characteristic: Optional[Characteristic] =
↳ None, **kwargs: Any,) -> None:
    """
    Initialise a :class:`Number` instance .
    Arguments
    -----
        characteristic: :class:`Characteristic` \
            <galactic.characteristics.Characteristic>`
            The embedded characteristic.
    """
    if characteristic is None:
        super().__init__(**kwargs)
    else:
        super().__init__(characteristic, **kwargs)

def __str__(self) -> str:
    if len(self.components) == 0:
        return "~"
    return str(self.components[0])

@classmethod
def constructor(cls, value: Any) -> Optional[float]:
    """
    Transform a value.
    Arguments
    -----
        value:
            The value to transform
    Returns
    -----
        float, optional
            The transformed value
    """
    if value is None:
        return None
    try:
        value = float(value)
        if math.isnan(value):
            return None
        return value # type: ignore
    except (TypeError, ValueError):
        return None

```

J'ai gardé les différentes méthodes de la classe `Number` telles qu'elles sont. Je n'ai modifié que la méthode `constructor`. Si la valeur renseignée en entrée, une fois transformée se situe entre 0 et 1, la méthode `constructor` renvoie un ensemble de valeurs possibles, sinon elle ne retourne rien. J'ai également écrit un fichier test afin d'effectuer un test pour chaque méthode de la classe `FuzzyValue`,

néanmoins Monsieur Demko m'a fait remarquer qu'il manquait des fichiers, dans mon dossier où se trouve mon code, afin que le plugin soit totalement opérationnel, il m'a conseillé de regarder l'ensemble des fichiers qui avaient été créés pour le plugin de caractéristiques numériques. En regardant les différents fichiers qu'il y avait, j'ai constaté qu'il manquait une dizaine de fichiers pour que le plugin soit complet. Voici les différents fichiers supplémentaires qu'il fallait créer:

- README.rst, qui est une description du plugin et de ce qu'il fait.
- CODE_OF_CONDUCT, qui est le texte de bonne conduite des contributeurs.
- LICENSE, qui est la licence de copyright.
- MANIFEST.in, qui précise tout ce qu'il faut inclure comme fichier.
- setup.cfg, précise le nom du plugin ainsi que la version sur laquelle il est disponible.
- setup.py, précise la version des différents modules et de python nécessaires à l'utilisateur pour utiliser le plugin.
- api.rst qui se trouve dans le dossier docs.
- conf.py, se trouvant dans le dossier docs, précise les modules nécessaires au bon fonctionnement du plugin.
- description.rst, se situant dans le dossier docs, donne une description un peu plus détaillée du plugin que le README.rst.
- index.rst se trouve dans le dossier docs.
- tox.ini, il s'agit d'un fichier utiliser pour faire passer différents tests au plugin. Ci-dessous on peut voir les différentes commandes qui seront lancées pour les tests.

```
[tox] envlist = py39,py38,py37,py36 #envlist = py38
```

```
[testenv] install_command = pip install -no-cache-dir -upgrade -upgrade-strategy eager  
-find-links https://galactic.univ-lr.fr/packages {opts} {packages} deps = .[test] commands =  
py38: doc8 -q docs py38: mypy -package galactic_characteristic_fuzzy py38: black -check  
-diff galactic_characteristic_fuzzy tests docs setup.py py38: pylint -rcfile=setup.cfg galac-  
tic_characteristic_fuzzy
```

```
pytest --cov=galactic_characteristic_fuzzy  
pytest --doctest-modules galactic_characteristic_fuzzy
```

- envlist permet de spécifier sur quelles versions de python les test doivent être effectués.
- install_command permet de récupérer les packages de la plateforme GALACTIC.
- mypy -package galactic_characteristic_fuzzy, permet de vérifier le code et de trouver des possibles erreurs dans le package galactic_characteristic_fuzzy.

- `doc8 -q docs` permet de vérifier que le code est correctement écrit.
- `black --check --diff` permet de vérifier le bon formatage des différents fichiers présents dans le dossier.
- `pylint --rcfile=setup.cfg galactic_characteristic_fuzzy` permet de vérifier que chaque ligne de code de chaque fichier du dossier ne dépasse pas un certain nombre de caractères par ligne. Ici, les lignes ne doivent pas dépasser 88 caractères.
- `pytest --cov=galactic_characteristic_fuzzy`, permet de produire un rapport de couverture. La couverture de tests est utilisée afin de décrire la mesure dans laquelle le code source d'un programme est exécuté lorsqu'une suite de test est exécutée.
- `pytest --doctest-modules galactic_characteristic_fuzzy` permet de vérifier les doctest, cela correspond au ligne de code commençant par `>>>`. Cela permet de faire une comparaison entre ce que renvoie réellement notre code et ce que l'on attend comme retour.

J'ai exécuté les différents tests pour le plugin sur mon terminal en lançant la commande `tox`. J'ai fait passer les tests un par un d'abord afin de corriger les erreurs au fur et à mesure, et ensuite j'ai fait passé tous les tests d'un coup avant de faire un `git push` sur ma branche de travail sur GitLab afin de vérifier que tous les tests sur GitLab passaient, ce qui ne fut le cas qu'au bout de quelques jours. Ne trouvant pas où était le problème Monsieur Demko faisait des commentaires à travers ma branche de travail sur GitLab afin que je puisse corriger toutes les erreurs pour pouvoir passer tous les tests. Une fois chose faite tous les tests sont passés et j'ai fait une merge request pour que ma branche soit fusionner avec la branche principale.

6.3 Mise en place du plugin de description



Figure 8: Logo représentant les plugins de description de la plateforme GALACTIC.

Un plugin de description nous permet de décrire les données extraites dans notre cas il s'agira de décrire des données floues.

Pour mettre en place le plugin de description des données floues je me suis inspirée du plugin de description numérique. Contrairement à ce dernier qui a été établi dans un demi-plan afin de pouvoir décrire les données numériques qui se situent dans $\mathbb{R}^2, \mathbb{R}^3$, la description des données floues

ne nécessite pas de demi-plan de ce fait il est plus simple à mettre en place. En effet, on sait que dans la logique floue l'ensembles des valeurs se situent entre 0 et 1 donc on peut utiliser un simple plan.

Il nous faut établir des prédicats qui nous permettront par la suite de mettre en place une description des données floues, ainsi nous pourrons trouver l'enveloppe convexe de l'ensemble des données traitées. Comme avec la logique floue nous sommes dans l'ensemble des nombres réels trouver l'enveloppe convexe revient à trouver la plus petite et la plus grande valeur de l'ensemble des données traitées. Du coup pour la création du plugin de description, on peut s'inspirer du plugin de description numérique en prenant en compte qu'il a été construit dans un demi-plan alors que le plugin de description des données floues est dans un simple plan. Pour mettre en place le plugin on peut s'appuyer sur la Théorie des Possibilités.

La Théorie des possibilités:

Elle permet de raisonner sur des connaissances imprécises ou vagues. On parle de possibilité et de nécessité. La possibilité va nous permettre d'évaluer quelle est l'une des valeurs possibles de la valeur qu'on a extraite. La nécessité quant à elle va nous permettre d'évaluer quelle est la valeur nécessaire à la valeur extraite, ainsi la valeur nécessaire correspond à la valeur minimale de l'enveloppe convexe que l'on cherche pour le plugin de description.

6.4 Mise en place du plugin de stratégie



Figure 9: Logo représentant les plugins de stratégie de la plateforme GALACTIC.

Un plugin de stratégie nous permet de décomposer les données en plusieurs sous-groupes.

D'une manière générale nous allons tout simplement partager l'ensemble en plusieurs quantiles, on peut diviser en trois, quatre, cinq quantiles ou même plus. Il faut donc réduire l'ensemble des données afin de pouvoir mettre en place plusieurs sous-groupes. Prenons un exemple, nous savons que l'ensemble des valeurs extraites seront entre 0 et 1, partageons cet ensemble en cinq quantiles équitables. Cela donnerait :

- Premier quantile : 0.0 → 0.2
- Deuxième quantile : 0.2 → 0.4

- Troisième quantile : 0.4 → 0.6
- Quatrième quantile : 0.6 → 0.8
- Cinquième quantile : 0.8 → 1

Prenons ensuite ensemble de données simple : [0.65 0.70 0.75 0.80 0.85], et supprimons la plus petite et la plus grande valeur, donc 0.65 et 0.85, en se posant la question suivante : à partir de quel seuil, dans les quantiles, pouvons-nous supprimer la valeur ? Pour pouvoir supprimer la valeur 0.65 du groupe il faut atteindre le seuil de 0.80 ainsi pour le premier groupe nous prendrons les valeurs supérieures ou égales à 0.80 (voir schéma ci-dessous). Et pour supprimer la valeur 0.85 du groupe il faut également atteindre le seuil de 0.80 ainsi nous prendrons les valeurs inférieures ou égales à 0.80 (voir schéma ci-dessous). Cela revient plus ou moins à utiliser les fonctions d'appartenances expliquées dans la présentation de la logique floue.



Figure 10: Exemple de stratégie

Pour la mise en place du plugin de stratégie, il faut clairement définir cette dernière, c'est à dire déterminer en combien de quantiles nous allons séparer l'ensemble allant de 0 à 1 inclus.

6.5 Le travail réalisé

J'ai mis en place le plugin de caractéristiques pour les données floues avec l'ensemble des fichiers nécessaires pour que tous les tests sur le GitLab du projet GALACTIC passent et que le plugin puisse être opérationnel pour être utilisé sur la plateforme. J'ai également commencé à mettre en place le plugin de description.

6.6 Le travail effectué

Il me reste à écrire les tests pour le plugin de description ainsi qu'à tous les faire passer sur ma machine mais aussi sur le GitLab du Projet et il me reste encore à mettre en place tout le plugin de stratégie.

7 Les outils utilisés durant mon Stage

- Pycharm pour écrire les différents plugins et leurs fichiers tests respectifs.
- Plateforme GALACTIC pour pouvoir utiliser les modules du Projet.
- Système d'exploitation Ubuntu pour faire l'installation de la plateforme.
- GitLab afin de pouvoir rajouter le code sur le projet GALACTIC.

- Visual Studio Code pour la rédaction de mon rapport de stage ainsi que pour le diaporama de présentation pour ma soutenance de stage.

8 Conclusion

J'ai effectué mon stage au sein de l'équipe Modèles et Connaissances du L3i, plus précisément au sein de l'équipe du projet GALACTIC. Il m'a été demandé de mettre au point trois nouveaux plugins pour la plateforme, permettant de gérer les données floues. Un plugin de caractéristiques, donc un plugin permettant d'extraire les données souhaitées d'un individu; un plugin de description qui établit un ensemble de prédicats nous permettant d'établir une description des données et pour finir un plugin de stratégie nous permettant de séparer les données en plusieurs sous-groupes grâce à la stratégie qui aura été mise en place.

Au jours d'aujourd'hui j'ai réussi à créer entièrement le plugin de caractéristiques pour les données floues, avec tous les tests nécessaires au bon fonctionnement du plugin et pour qu'il soit totalement opérationnel, il a été rajouté aux différents plugins déjà opérationnels sur la plateforme GALACTIC. Concernant les deux autres plugins, que je devais mettre je n'ai malheureusement pas fini de les mettre au point, sur tout par manque de temps. En effet, l'installation fut l'un des problèmes majeures durant mon stage malheureusement elle m'aura pris une semaine, mais il y a également toute les recherches concernant la logique floue ainsi que la Théorie des possibilités qui m'ont pris plus de temps que je ne le pensais, en effet il fallait bien cerner sur le sujet sur la logique floue avant de commencer à mettre en place les différents plugins qui m'étaient demandés. Il m'a également fallu comprendre comment les plugins de caractéristiques et de description ont été établis étant donné que je devais m'appuyer dessus pour la création des plugins de caractéristiques et de description des données floues. Néanmoins, il est prévu que je finisse la mise en place des deux plugins manquants, même après la fin de mon stage, en effet j'ai demandé à Monsieur Demko si il m'était possible de finir les plugins après la fin de mon stage et j'ai eu son accord.

Ce stage m'a donné l'occasion de découvrir et d'apprendre de nouvelles choses et d'en apprendre plus sur moi-même. En effet, il m'a permis de constater les lacunes que j'ai en informatique. Mais il m'a également permis de voir comment se dérouler le travail dans un projet d'équipe et comment se dérouler les réunions d'équipes, j'ai également pu apprendre comment fonctionner le travail de recherche dans un laboratoire Informatique. Ce stage m'a aidé à prendre conscience qu'il faut un certain nombre de fichiers pour qu'un plugin puisse être totalement opérationnel. Il m'a également permis d'apprendre à rédiger un rapport de stage et un diaporama de présentation en Markdown.

9 Annexe - Sources

Présentation de l'entreprise

<https://l3i.univ-larochelle.fr/Presentation-317> <https://l3i.univ-larochelle.fr/Equipes> <https://www.univ-larochelle.fr/wp-content/uploads/pdf/ULR-FICHES-LABO-L3i-2020-vf-web.pdf>

Présentation de la Plateforme GALACTIC

<https://galactic.univ-lr.fr/slides/architecture/Galactic-Architecture-slides-20min-complete.pdf>
<https://galactic.univ-lr.fr/slides/architecture/Galactic-Architecture-slides-20min-with-notes.pdf>
<https://galactic.univ-lr.fr/slides/fca/Galactic-FCA-slides-with-notes.pdf>

La logique floue et la Théorie des Possibilités

<https://web4.ensiie.fr/~gacogne/introflou.pdf>

<https://eduscol.education.fr/sti/sites/eduscol.education.fr/sti/files/ressources/pedagogiques/598/598-presentation-logique-floue.doc>

(Si le lien au-dessus ne fonctionne pas: saisir dans la barre de recherche “Logique floue eduscol” et cliquez sur le premier lien, c’est un document à télécharger.)

https://fr.wikipedia.org/wiki/Théorie_des_possibilités <http://departements.imt-atlantique.fr/data/sc/seminaires/Possibilite%CC%81.pdf>