

Master Informatique

La Rochelle Université

GALACTIC - EXPLORER



Romain Coste

2024

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

Table des matières

1 Remerciements	4
2 Introduction	4
2.1 Présentation de l'organisme d'accueil	4
2.2 GALACTIC-Explorer	5
2.3 Thèmes et étapes du projet	6
2.4 Planning prévisionnel	6
2.5 Choix techniques	7
2.5.1 GALACTIC	7
2.5.2 Python	9
2.5.3 MSYS2	10
2.5.4 GTK3	11
2.5.5 Poetry	11
3 Travail réalisé	12
3.1 Prise en main du sujet et de l'interface GALACTIC-Explorer	12
3.2 Réunion (8 mars)	12
3.3 Amélioration de l'ergonomie et fonctionnalités de l'interface	16
3.4 Plug-ins	17
3.5 Ajout fenêtres secondaires	19
3.5.1 Fenêtre d'introduction	19
3.5.2 Fenêtre d'accueil	20
3.6 Préférences utilisateur	21
3.7 Gestion de l'application	22
3.8 Nettoyage + restructuration du code	23
4 Travail en cours + prévision	24
5 Difficultés rencontrés	26
6 Activités au sein de l'entreprise	26
7 Conclusion	27
8 Conclusion (english)	28
9 Apport du stage/Projet futur	29
9.1 Apport du stage	29
9.2 Projet futur	29
10 Sitographie	29

Liste des figures

1	Logo de l'entreprise	4
2	Logo de GALACTIC-Explorer	5
3	Croquis de l'interface	6
4	Portrait de Evariste (réalisé par M.GAUTREAU)	7
5	Fleur de GALACTIC	8
6	Logo Python	9
7	Logo MSYS2	10
8	Logo GTK	11
9	Logo Poetry	11
10	Croquis de page d'accueil de GALACTIC-Explorer	13
11	Croquis de page de présentation	14
12	Menus de l'interface	14
13	Barre d'outils de l'interface	15
14	Aperçu menu File	15
15	Aperçu menu Edit	15
16	Aperçu menu Workspace	15
17	Pictogrammes de l'interface	16
18	Fenêtre pop-up pour l'exportation	18
19	Page de présentation GALACTIC-Explorer	20
20	Page d'accueil GALACTIC-Explorer	21
21	Fenêtre préférence utilisateur	22
22	Fenêtre de sauvegarde avant fermeture	23
23	Fenêtre de sauvegarde avant changement de fenêtre	23
24	Aperçu interface principale	24

Résumé

Lors de ce stage, qui s'est déroulé du 5 février au 21 juin 2024, j'ai travaillé sur le sujet proposé par M. Christophe Demko : Ergonomie d'une interface logicielle GALACTIC-Explorer. Le projet est financé par la SATT (Société d'Accélération de Transfert Technologique) de la Nouvelle-Aquitaine. L'objectif principal de cette interface est de permettre à un utilisateur d'effectuer des analyses interactives de données complexes et hétérogènes. Il aura la possibilité d'interagir avec les données et leur sémantique, pour une détection interactive de corrélations qui peuvent décrire des comportements, spécificités et anomalies présentes seulement dans certaines parties des données. Mon travail au sein de ce projet est d'apporter diverses améliorations à partir du travail réalisé par le développeur (Étienne Descamps) qui a fourni une interface fonctionnelle basique, mais complète. Des améliorations aussi bien sur l'aspect visuel de l'interface que des améliorations afin de faciliter le parcours utilisateur pour guider pas à pas les novices ou rendre l'interface fluide pour des utilisateurs aguerris.

During this internship, which took place from February 5th to June 21st, 2024, I worked on the subject proposed by Mr. Christophe Demko: Ergonomics of a GALACTIC-Explorer software interface. The project is funded by the SATT (Société d'Accélération de Transfert Technologique) of Nouvelle-Aquitaine. The main objective of this interface is to allow a user to perform interactive analyses of complex and heterogeneous data. The user will have the ability to interact with the data and its semantics, for an interactive detection of correlations that can describe behaviors, specificities, and anomalies present only in certain parts of the data. My work within this project is to bring various improvements based on the work done by the developer (Étienne Descamps) who provided a basic but complete functional interface. Improvements include both the visual aspect of the interface and enhancements to facilitate user navigation to guide novices step by step or to make the interface smooth for experienced users.

1 Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude envers toute l'équipe pédagogique de La Rochelle Université qui m'a permis d'acquérir les compétences théoriques nécessaires pour mener à bien ce stage.

Ensuite, je souhaite remercier chaleureusement l'ensemble de l'équipe du Laboratoire Informatique, Image et Interaction (L3i) qui m'a accueilli dans ses locaux et m'a mis à disposition une machine fonctionnelle, me permettant ainsi de travailler dans des conditions optimales pendant mon stage.

Enfin, j'aimerais exprimer ma sincère gratitude envers Monsieur Christophe Demko, mon maître de stage, ainsi que Madame Karell Bertet, ma tutrice de stage, pour leur précieuse aide, leurs conseils avisés et l'opportunité exceptionnelle de travailler sur ce sujet de stage au sein du L3i.

2 Introduction

2.1 Présentation de l'organisme d'accueil



Figure 1: Logo de l'entreprise

(cette présentation est tirée de mon rapport de stage lors de ma première année de Master)

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 en 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 partie des réseaux de recherche 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 volontariste de participation ou de pilotage de projets de recherche labellisés (ANR, PCRD, etc.), 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 recherche 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 L3i est divisé en trois 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-Explorer de cette équipe.

2.2 GALACTIC-Explorer



Figure 2: Logo de GALACTIC-Explorer

Des résultats récents et novateurs obtenus au laboratoire L3i de La Rochelle Université permettent d'envisager de proposer de nouvelles méthodes XIA d'analyse de données complexes et hétérogènes offrant à la fois explicabilité et interactivité. Il s'agit d'un nouvel algorithme, NextPriorityConcept1, qui résout deux verrous scientifiques dans les domaines de l'analyse de données et de l'analyse formelle des concepts, ainsi que de la plateforme GALACTIC qui en fournit une implémentation générique et extensive grâce à un mécanisme de plugin pour un ajout facilité de nouveaux types de données et de stratégies d'exploration.

Dans le cadre d'un financement SATT (Société d'Accélération de Transfert Technologique) de la Nouvelle-Aquitaine, je travaille actuellement sur le logiciel GALACTIC-Explorer, qui repose sur une montée en maturation de la plateforme GALACTIC. L'objectif de ce logiciel est de proposer une analyse interactive de données complexes et hétérogènes guidée par l'expert qui, positionné au cœur du processus d'analyse, aura la possibilité d'interagir avec les données et leur sémantique pour une détection interactive de corrélations pouvant décrire des comportements, spécificités et anomalies présentes seulement dans certaines parties des données.

Une interface basique mais complète des traitements de GALACTIC-Explorer a été développée. L'objectif de ce stage sera de l'améliorer dans une perspective de commercialisation, avec des éléments graphiques proposés par un ergonomiste logiciel.

2.3 Thèmes et étapes du projet

- Ajout de nouvelles fonctionnalités pour rendre l'utilisation et la navigation de l'interface plus faciles et intuitives pour les utilisateurs novices comme expérimentés.
- Travail en collaboration avec un ergonomiste (M. Gautreau) pour améliorer l'ergonomie de l'interface et adopter une charte graphique pour l'application GALACTIC-Explorer (couleurs, logos, pictogrammes).
- Mise en place d'un système de plugins pour l'application afin d'ajouter de nouvelles fonctionnalités (exportation, visualisations, etc.) et permettre ainsi aux utilisateurs aguerris de créer leurs propres sous-plugins (ex : ajouter un autre type d'extension pour l'exportation).
- Autres axes d'amélioration selon l'avancée du travail.

2.4 Planning prévisionnel

Un planning prévisionnel n'a pas vraiment été mis en place. L'ancien développeur de l'interface a laissé différentes tâches à réaliser dans l'ordre que je souhaite. En mars, j'ai eu une réunion avec mon maître de stage (M. Demko) et un ergonomiste (M. Gautreau) afin de définir une première version de l'interface avec un schéma clair, aussi bien sur le côté visuel que sur le côté pratique. Durant cette réunion, il a également été précisé la création d'un thème GTK propre à l'interface et la mise en place d'un système de plugins.

Il n'y a pas vraiment d'ordre précis pour réaliser ce projet, l'objectif final étant de rendre une première version fonctionnelle avec tous les points abordés. Bien entendu, si cette première version est aboutie avant la fin de mon stage, il sera possible d'ajouter des éléments d'ergonomie ou d'autres fonctionnalités pour améliorer l'application.

Ci-dessous, le croquis réalisé par M. Gautreau suite à notre réunion :

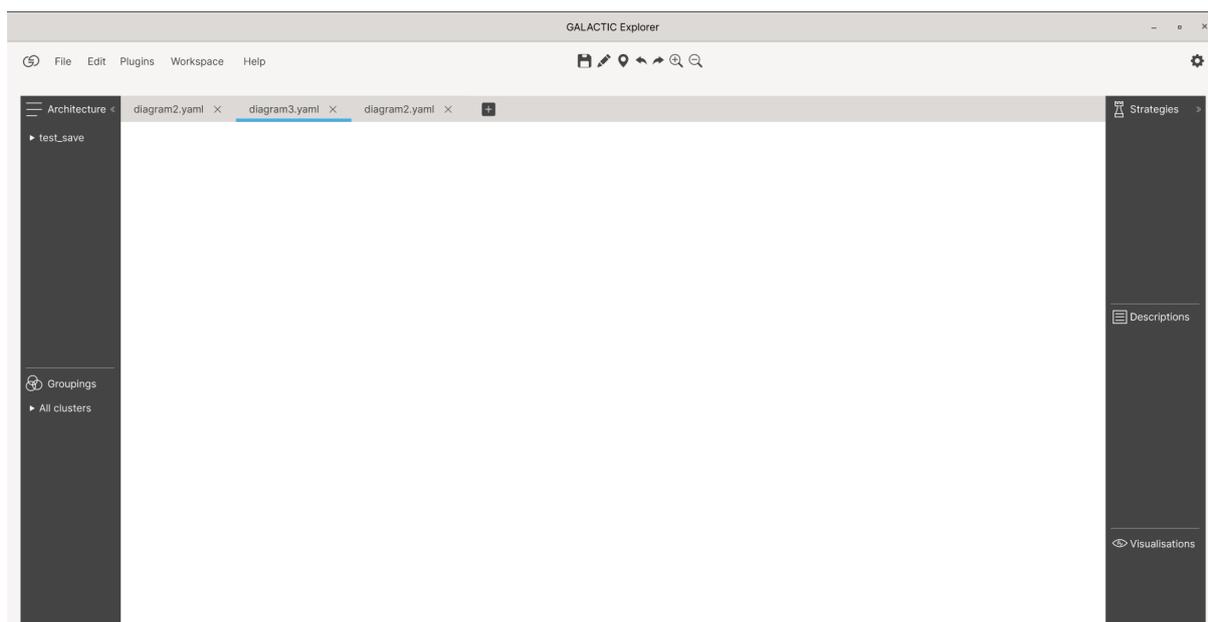


Figure 3: Croquis de l'interface

2.5 Choix techniques

Pour ce projet, je n'ai pas eu besoin de faire de choix techniques, étant donné que j'ai repris les travaux d'un ancien développeur. Je vais donc présenter brièvement les différents outils techniques que j'ai utilisés lors de mon stage :

2.5.1 GALACTIC

GALACTIC est l'acronyme de "GALois LATTices, Concept Theory, Implicational Systems and Closures" (référence au mathématicien français Évariste Galois). Le but de GALACTIC est de développer un framework basé sur la théorie des treillis et l'analyse formelle de concepts (basé sur les travaux de Bernhard Ganter et Rudolf Wille, ainsi que de Marc Barbut et Bernard Monjarnet).



Figure 4: Portrait de Evariste (réalisé par M.GAUTREAU)

GALACTIC est une bibliothèque écrite en Python avec une architecture dite de "fleur". Le cœur de cette bibliothèque est générique dans le sens où l'on peut utiliser n'importe quel type de données. Afin de travailler sur différents types de données, le cœur va avoir besoin de différents types d'extensions :

- des applications
- des plugins de caractéristiques qui vont permettre de gérer différents types de données
- des plugins de descriptions qui vont décrire les différents types de données utilisées
- des plugins de stratégies qui vont fournir des stratégies d'exploration pour les différents types de données
- des plugins de mesures qui vont mesurer les différents types de données
- des plugins de lecteur de données afin de gérer différents flux de données
- des plugins de localisation qui vont permettre une traduction automatique des différentes applications

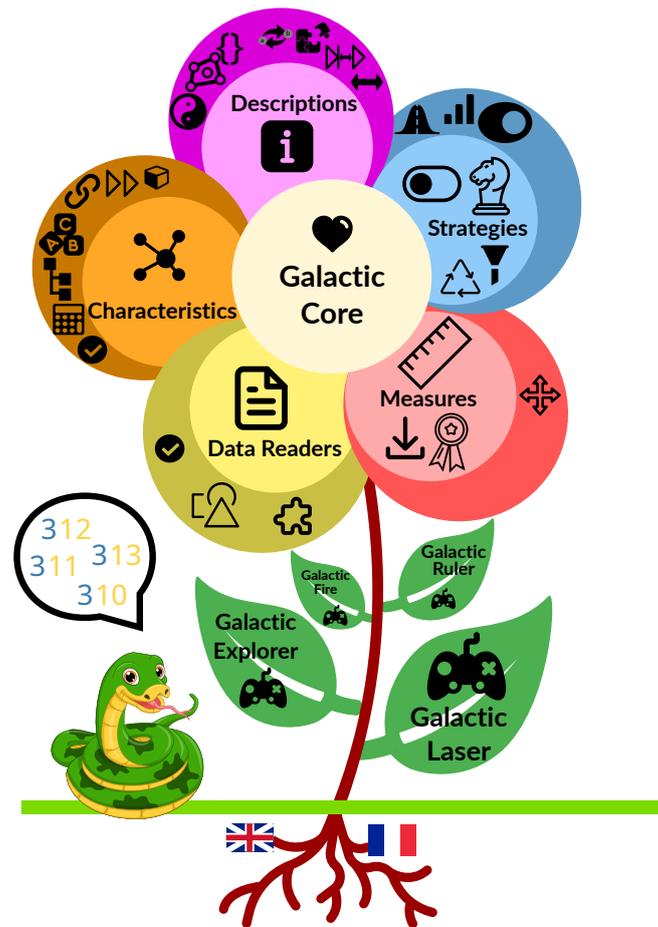


Figure 5: Fleur de GALACTIC

Avec le cœur de la bibliothèque ainsi que l'ensemble de ces plugins, les données vont pouvoir être traitées selon un certain processus :

- Les données vont d'abord être lues par les plugins de type lecteur afin d'obtenir un nuage de points.
- Suite à ce nuage de points, on va extraire des caractéristiques avec le plugin correspondant.
- On va ajouter une description à ce nuage de points, autrement dit, on va ajouter une enveloppe convexe qui est décrite par les demi-plans qui composent les faces de cette enveloppe convexe.
- Cette enveloppe convexe va ensuite être divisée en plusieurs parties grâce aux plugins de stratégies.

Une fois cette phase de préparation des données terminée, on va utiliser l'algorithme "Next Priority Concept" qui va permettre de définir des concepts. Ensuite, on pourra construire un treillis (structure algébrique mathématique).

Le cœur de GALACTIC va contenir les opérations de base, les structures de données de base et l'algorithme "NextPriorityConcept".

Les plugins de caractéristiques définissent de nouvelles façons de gérer les données (données de type booléen, numérique, catégoriel, chaîne de caractères, etc.).

Les plugins de descriptions vont permettre de définir l'enveloppe convexe généralisée des données par l'intermédiaire de demi-plans. Autrement dit, les descripteurs sont des prédicats monadiques (à argument unique) afin de déterminer si tel objet a ou non la propriété de ce descripteur.

Les plugins de stratégies vont permettre de découper un concept en plusieurs parties avec potentiellement des découpes qui auront des parties communes. Au sein même des stratégies, il existe des méta-stratégies qui vont pouvoir se fixer aux stratégies déjà existantes pour limiter la profondeur d'exploration des données analysées.

- La méta-stratégie de type limit qui va garder uniquement les prédécesseurs dont la mesure n'excède pas la limite imposée.
- La méta-stratégie de type sélections qui va garder seulement les meilleurs ou les pires prédécesseurs.

Pour que les plugins de stratégies fonctionnent, ils utilisent les plugins de mesures qui sont définis dans le cœur (cardinalité du prédécesseur, cardinalité du successeur, la confiance).

Les applications liées à GALACTIC sont développées pour utiliser la librairie, ce sont en quelque sorte les interfaces pour les utilisateurs :

- GALACTIC-Laser : application qui va permettre de dessiner le diagramme de Hesses d'un treillis (disponible en ligne de commande).
- GALACTIC-Explorer : application qui va permettre l'interaction avec le treillis construit.
- GALACTIC-Ruler : application permettant d'extraire des règles d'implications à partir d'un treillis.
- GALACTIC-Fire : application permettant d'exécuter un système de règles.

2.5.2 Python



Figure 6: Logo Python

Python est un langage de programmation interprété, multiparadigme et multiplateforme. Ce langage contient un typage dynamique fort ainsi qu'une gestion automatique de la mémoire, et également un système de gestion des exceptions.

L'un des points forts du langage Python est qu'il fonctionne sur une multitude de plateformes informatiques, des smartphones (Android, iOS) aux systèmes d'exploitation divers tels que Windows, Unix et macOS. Il peut également être traduit en Java ou .NET.

Python est également très utile pour l'apprentissage selon certains pédagogues. Grâce à sa syntaxe claire, séparée des mécanismes de bas niveau, il permet une initiation aisée aux concepts de base de la programmation.

Python est utilisable dans de nombreux contextes et s'adapte à tout type d'utilisation grâce à des bibliothèques spécialisées qui permettent d'ajouter une multitude de fonctionnalités utiles aux développeurs.

Mais son utilisation principale reste en tant que langage de script pour automatiser des tâches simples mais fastidieuses. Python est également utilisé en tant que langage de prototypage pour développer une application fonctionnelle avant de l'optimiser avec un langage de plus bas niveau.

Les principaux domaines d'utilisation de Python sont l'informatique quantique, l'intelligence artificielle, ainsi que le traitement des Big Data et le Machine Learning.

Dans le cadre du projet GALACTIC-Explorer, Python est le langage de programmation utilisé pour la réalisation de l'interface graphique.

2.5.3 MSYS2



Figure 7: Logo MSYS2

MSYS2 est une plateforme de distribution et de développement logiciels permettant d'utiliser des logiciels conçus pour Unix sur Windows.

MSYS2 comprend le gestionnaire de paquets Pacman d'Arch Linux, ce qui permet d'installer la plupart des paquets présents sur Linux directement sur le système d'exploitation Windows. Il contient également un shell Bash et d'autres programmes Unix.

Dans le cadre du projet GALACTIC-Explorer, le développeur initial de l'interface graphique (Etienne DESCHAMPS) utilise un environnement MSYS2 personnalisé avec une multitude de paquets installés afin de faire fonctionner GALACTIC-Explorer sur Windows. L'installation de l'interface n'est pour l'instant pas disponible directement sur Windows, car la bibliothèque utilisée en Python (GTK) ne fonctionne pas de façon optimale sur Windows.

2.5.4 GTK3

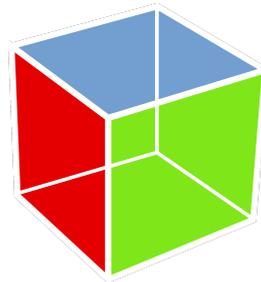


Figure 8: Logo GTK

GTK (the Gimp ToolKit) est une bibliothèque Python qui contient elle-même un ensemble de bibliothèques logicielles permettant la réalisation d'interfaces graphiques.

GTK est écrite en langage C, mais elle utilise le paradigme de la programmation orientée objet grâce à la bibliothèque GObject. GTK est utilisable dans une multitude de langages comme C++, JavaScript et Python.

GTK peut être assez complexe à prendre en main au départ en raison du fait que la majorité de la documentation soit pour la version C de cette bibliothèque. Une fois que GTK est maîtrisé, il permet de réaliser une multitude de choses afin d'obtenir une interface personnalisée et fonctionnelle.

En raison de la multitude des bibliothèques installées pour l'utilisation de GALACTIC-Explorer, son interface est réalisée avec GTK3. Il est pour l'instant impossible de passer à GTK4 en raison des différents problèmes de compatibilité liés à la liste des bibliothèques installées.

2.5.5 Poetry



Figure 9: Logo Poetry

Poetry va permettre de réaliser des projets Python complexes en déclarant, gérant et installant toutes les dépendances liées à un projet Python en particulier.

Dans un fichier spécifique (`pyproject.toml`), Poetry va répertorier toutes les dépendances nécessaires au fonctionnement d'un projet Python et va également gérer par lui-même la compatibilité des différentes dépendances afin de les faire fonctionner ensemble. Le créateur du projet peut passer par

Poetry pour ajouter ou supprimer des dépendances. Les utilisateurs n'auront plus qu'à effectuer une installation simple grâce à Poetry, qui va se charger d'installer toutes les dépendances.

Dans le cadre du projet GALACTIC-Explorer, Poetry est très utile, car vu la multitude de bibliothèques Python ainsi que toutes les dépendances liées à GALACTIC, il serait long et fastidieux de tout installer manuellement. Poetry me permet donc de construire le projet de l'interface et d'utiliser un certain type de fichier créé suite à la construction (.whl) afin de l'utiliser dans l'environnement MSYS2 pour exécuter l'application sous GTK sur Windows.

3 Travail réalisé

3.1 Prise en main du sujet et de l'interface GALACTIC-Explorer

La première semaine de mon stage s'est surtout centrée sur la découverte du sujet. Avec une interface déjà réalisée au préalable, il m'a fallu un certain temps pour comprendre le fonctionnement du code derrière cette interface, qui est à la fois complète et complexe. De plus, l'intégralité de l'interface est réalisée avec la bibliothèque GTK en Python, donc j'ai dû apprendre à la fois cette bibliothèque que je n'avais jamais utilisée et également le code de l'application.

Avant de passer à la phase d'apprentissage, j'ai d'abord installé l'interface pour vérifier si elle était fonctionnelle. L'installation a pris un certain temps, car GTK ne fonctionne pas très bien sous Windows. J'ai donc utilisé WSL (Windows Subsystem for Linux) pour compiler entièrement le code dans un fichier whl. Pour lancer l'application, l'ancien développeur a laissé un lien vers un environnement MSYS2 sur la page Git de ce projet. Cet environnement permet de lancer l'application à partir de Windows sans rencontrer de problèmes avec GTK.

Avec l'interface fonctionnelle devant moi, il était plus facile de comprendre le code qui la sous-tend.

Avant d'apporter la moindre modification au code, je me suis rendu sur la page GitLab du projet GALACTIC-Explorer (fourni par M. DEMKO) et j'ai ajouté une nouvelle branche en prenant comme base le code réalisé par l'ancien développeur, afin de pouvoir effectuer des modifications sans risquer de rendre l'application inutilisable.

3.2 Réunion (8 mars)

M. DEMKO m'a laissé un mois pour me familiariser avec l'interface GALACTIC-Explorer et la bibliothèque GTK en Python. Il a programmé une réunion avec un ergonomiste (M. GAUTREAU) afin de décider des différents axes d'amélioration pour rendre l'interface plus ergonomique et améliorer son visuel global. Durant cette réunion, de nombreux points ont été abordés, aussi bien sur les aspects pratiques que visuels, afin d'améliorer au maximum l'expérience utilisateur.

L'ajout d'une page d'accueil a été proposé pour faciliter la création ou l'accès à un projet pour un utilisateur novice.

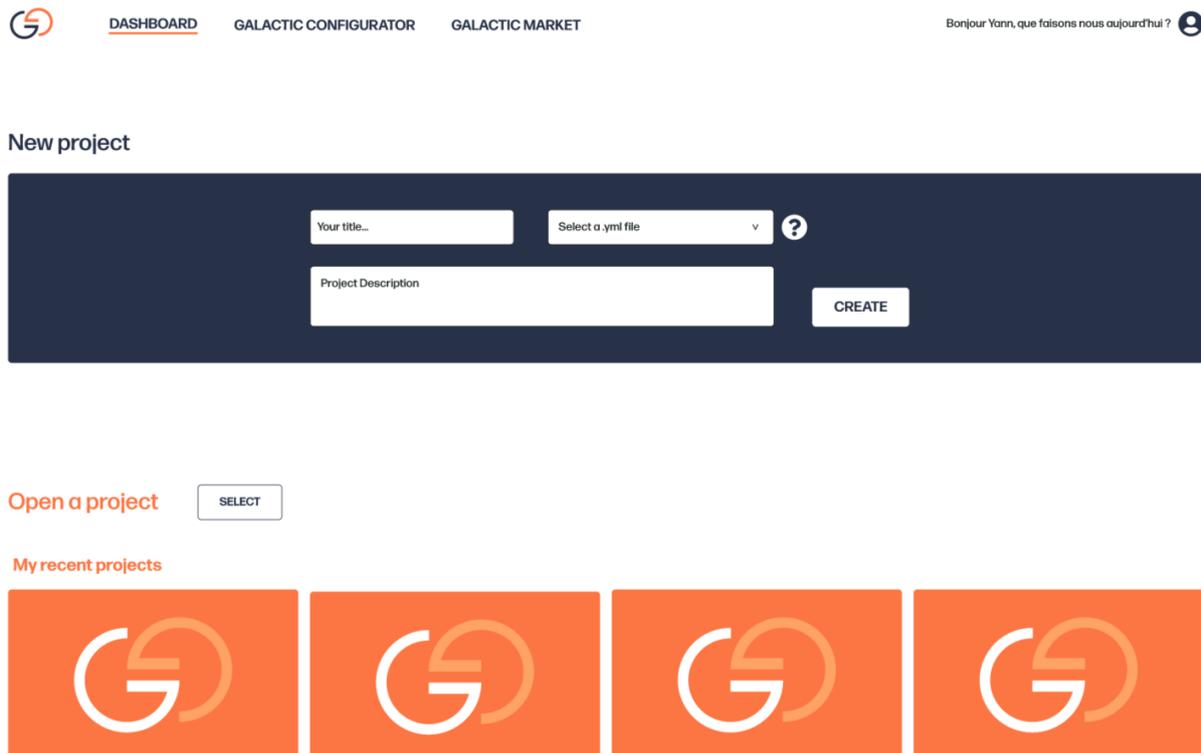


Figure 10: Croquis de page d'accueil de GALACTIC-Explorer

Ci-dessus, un croquis de la page d'accueil réalisé par M. GAUTREAU afin d'avoir une idée précise de ce qui est attendu pour le tableau de bord de GALACTIC-Explorer. On peut voir une première partie "New project" permettant à l'utilisateur d'instancier un projet avec ses propres données. Dans l'encadré, il est demandé un nom pour le projet, une description ainsi qu'un menu déroulant pour choisir un fichier de type YAML, car GALACTIC-Explorer travaille avec ce type de fichier pour fonctionner. Si l'utilisateur n'est pas à l'aise avec ce genre de fichier, le point d'interrogation à côté du menu déroulant lui donnera la marche à suivre, notamment en utilisant GALACTIC Configurator, qui permet de créer un fichier YAML à partir de données fournies.

On peut également voir en haut à droite du croquis que l'utilisateur sera reconnu à chaque nouvelle ouverture de l'interface. Un autre point important pour l'ergonomie de GALACTIC-Explorer est la nécessité de sauvegarder les préférences utilisateurs. Par exemple, la partie "Open a project" utilisera cette sauvegarde afin de garder en mémoire les derniers projets ouverts par l'utilisateur, l'empêchant ainsi de devoir les chercher dans l'arborescence de sa machine. Il aura toujours la possibilité d'ouvrir une arborescence si l'un de ses projets se trouve ailleurs que dans le dossier contenant l'ensemble de ses projets. La sauvegarde des préférences utilisateurs aura également d'autres fonctionnalités, comme le choix de désactiver ou non cette page d'accueil, car un utilisateur expérimenté n'aura peut-être plus envie de perdre du temps sur cette page d'accueil guidée et voudra directement se retrouver sur l'interface principale. De même pour la disposition de son espace de travail, la sauvegarde permettra de garder en mémoire la disposition de son interface lors de sa fermeture.

Pour ajouter une touche esthétique au logiciel, M. GAUTREAU m'a également fourni une page de présentation à afficher quelques secondes à l'ouverture de l'application.



Figure 11: Croquis de page de présentation

Avec le nom de l'application, une description pour présenter GALACTIC-Explorer ainsi qu'une illustration réalisée par M. GAUTREAU lui-même.

Des modifications au sein de l'interface principale ont également été prévues. En effet, les menus à droite et à gauche de la fenêtre graphique pourront soit être affichés, soit être cachés. L'utilisateur aura également le choix de rendre ces fenêtres flottantes.

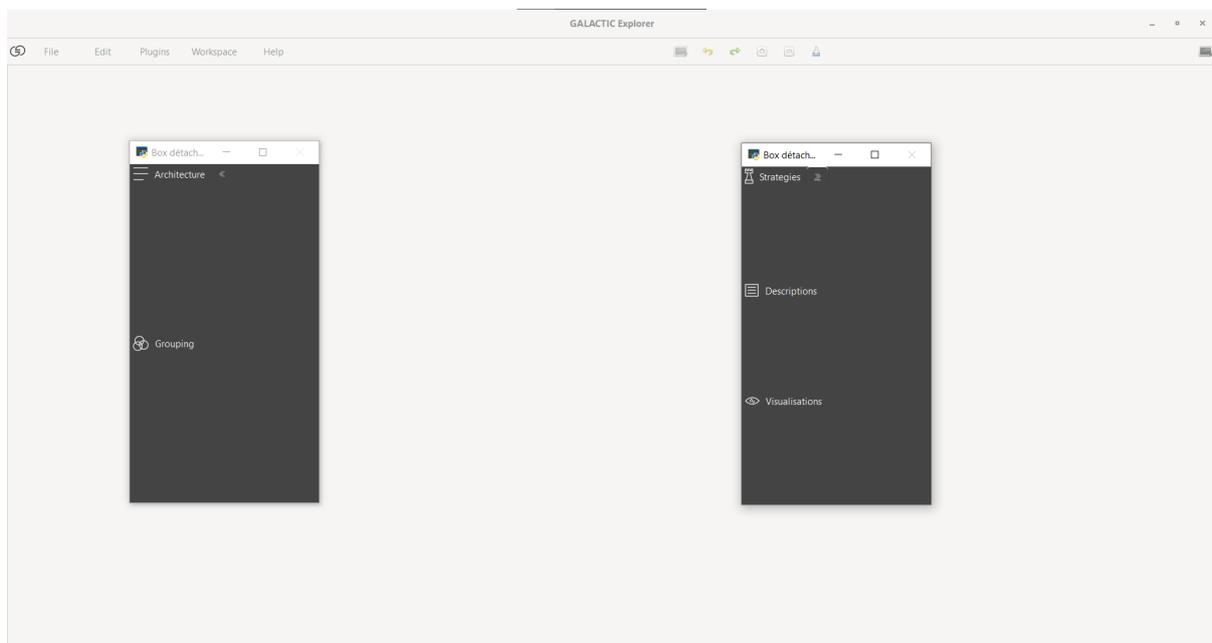


Figure 12: Menus de l'interface

Toutes ces options pourront être sauvegardées dans les préférences utilisateurs afin de conserver la

même disposition même après la fermeture de l'application.

La barre d'outils sera également modifiée. D'une part, il y aura un ensemble de menus avec des sous-menus pour les utilisateurs novices ; chaque bouton aura un nom précis pour guider l'utilisateur dans l'interface.



Figure 13: Barre d'outils de l'interface

Le menu "File" comprendra tous les outils pour gérer son projet : * Nouveau projet * Ouvrir un projet existant * Sauvegarder le travail actuel * Exporter le diagramme sous un certain format (plugins)

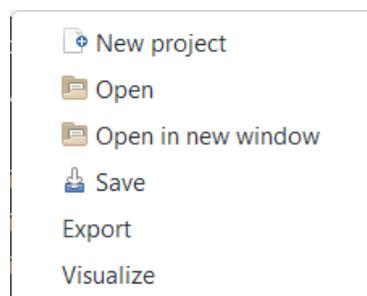


Figure 14: Aperçu menu File

Le menu "Edit" permettra, comme son nom l'indique, d'éditer le projet ouvert : * Retour arrière si l'utilisateur a fait une erreur * Retour avant si l'utilisateur veut revenir au point le plus récent * Nouveau diagramme afin de permettre à l'utilisateur d'ajouter un nouveau diagramme à son projet

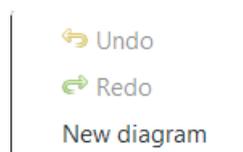


Figure 15: Aperçu menu Edit

Le menu "Plugins" permettra à l'utilisateur de gérer tous les plugins installés sur son application.

Le menu "Workspace" permettra de modifier son espace de travail à sa guise : * Afficher ou masquer les menus * Rendre les menus flottants * Permettre le zoom dans la zone graphique

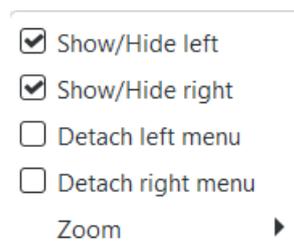


Figure 16: Aperçu menu Workspace

Le menu “Help” offrira plus d’explications sur certains points de l’interface.

Une autre partie de la barre d’outils sera agrémentée de pictogrammes pour accéder aux fonctionnalités beaucoup plus rapidement que par la navigation dans les menus.



Figure 17: Pictogrammes de l’interface

- Un pictogramme pour sauvegarder son travail
- Un pictogramme pour éditer le diagramme ouvert dans la fenêtre graphique
- Un pictogramme pour faire apparaître une mini-carte afin de se retrouver plus facilement dans un diagramme conséquent
- Deux pictogrammes pour modifier l’état du diagramme
- Deux pictogrammes pour gérer le zoom dans la fenêtre graphique du diagramme

3.3 Amélioration de l’ergonomie et fonctionnalités de l’interface

L’ancien développeur a laissé des commentaires dans son code pour m’aiguiller sur les axes d’amélioration à apporter pour ce projet. Certains étaient plus faciles que d’autres et m’ont permis de prendre en main GTK avec des exemples concrets.

Avec la documentation de GTK et le code déjà fourni, j’ai donc commencé à réaliser différentes tâches pour améliorer l’interface.

La première catégorie de tâches à effectuer pour améliorer l’interface est l’ajout de différents boutons. Certes, l’interface est déjà fonctionnelle en tant que telle, mais l’ajout de certains boutons pourrait rendre la navigation plus fluide. Avec mes modifications, il est maintenant possible d’ajouter un nouveau diagramme dans un projet existant, de cacher ou d’afficher les menus présents sur l’interface, de pouvoir modifier le projet dans son ensemble avec la possibilité de renommer un diagramme, de supprimer un fichier, de dupliquer un diagramme existant, et même d’unifier deux diagrammes. Toutes ces tâches suivent la même méthodologie. Pour les réaliser, il suffit d’utiliser un objet GTK (GTK.Button). Il faut ensuite le connecter à la méthode de son choix qui sera appelée à chaque fois que ce bouton sera pressé. Il faut ensuite placer les boutons dans l’interface. GTK utilise des BOX pour modéliser son interface comme on le souhaite. Il y a une interface principale et on ajoute ensuite tous nos éléments dans un ordre précis pour obtenir un visuel qui nous convient. Dans le cas de mes boutons, je les ai ajoutés dans une barre d’outils située en haut de l’interface. Une autre particularité des boutons est leur état dans l’interface. Par exemple, le bouton de sauvegarde sera désactivé tant qu’un projet n’est pas ouvert dans l’application. La plupart des boutons fonctionnent de la même façon : ils sont désactivés au lancement de l’interface et deviennent actifs lorsqu’un projet est créé ou chargé.

Une autre catégorie de tâches est la gestion des raccourcis clavier pour la sauvegarde du projet et le retour arrière/avant directement sur les diagrammes. J’ai mis en place une méthode qui écoute les actions de l’utilisateur ; si l’un des raccourcis est reconnu, la méthode va appeler l’une des méthodes reliées à ce raccourci clavier. Une autre gestion de raccourci m’a donné un peu plus de difficulté : le

double-clic pour renommer un cluster au sein d'un diagramme. Les clusters sont stockés dans un objet GTK (Gtk.Treeview) et ce type d'objet permet de sélectionner un élément particulier (donc de réagir à un simple clic). J'ai donc dû trouver une alternative pour permettre au double-clic d'activer une méthode pour renommer le cluster.

Suite à la réunion avec M. Gautreau, j'ai ajouté la page de présentation de GALACTIC-Explorer. J'ai modifié l'instanciation de la fenêtre principale en affichant d'abord la page de présentation puis, au bout de 5 secondes, celle-ci se ferme pour appeler la fenêtre principale. Je me suis aidé de la charte graphique fournie par M. Gautreau pour respecter les bonnes couleurs.

J'ai également fait plusieurs modifications à l'interface principale :

- Changer le nom de certains menus dans la barre d'outils.
- Rendre les menus flottants, permettant ainsi de les glisser dans de nouvelles fenêtres et de les fermer si l'utilisateur choisit de remettre les menus dans l'interface.
- Ajouter un système de zoom pour la fenêtre graphique avec un zoom avant/arrière pour les pictogrammes et un zoom plus précis dans le menu "Workspace" qui offre une liste déroulante de pourcentages, évitant ainsi de devoir cliquer plusieurs fois sur le pictogramme en cas de dézoom ou zoom important.
- Ajouter des boutons et des menus fictifs pour agrémenter l'interface. Les fonctionnalités liées à ces boutons seront développées ultérieurement.
- Mettre en place un fichier CSS pour modifier plus facilement l'aspect visuel de l'interface. GTK permet d'utiliser le langage CSS pour modifier les éléments visuels de ses objets.

3.4 Plugins

Avant de m'attaquer à cette partie, j'ai fait un point avec M. DEMKO pour m'assurer de partir dans la bonne direction et éviter de perdre des journées de travail. Lors de cette discussion, M. DEMKO m'a expliqué plus en détail le fonctionnement des plugins et des paquets en Python. Comme le projet GALACTIC-Explorer est créé avec Poetry (qui permet de gérer un projet Python), l'implémentation de plugins pour l'application sera simplifiée. Chaque type de plugin doit avoir un point d'entrée afin qu'il soit reconnu comme tel lorsque l'application recherche dans les différents paquets Python. Il est également impératif de préciser la méthode à utiliser pour l'exportation dans le fichier init du plugin afin qu'il soit lu par l'application.

Le premier type de plugin que je devrai implémenter est le plugin d'exportation. Celui-ci permettra d'exporter les diagrammes d'un projet dans un certain format, et les utilisateurs avancés pourront ajouter un sous-plugin d'exportation avec le type de fichier de leur choix. Ce premier type de plugin est assez simple à mettre en place, car la méthodologie restera relativement la même, seule l'extension du fichier changera.

Le deuxième type de plugin sera plus complexe. En effet, le plugin de visualisation permettra à l'utilisateur de visualiser soit un cluster seul, soit un groupe de clusters. Il y aura donc deux sous-catégories de plugins selon le nombre de clusters. Une autre difficulté de ce plugin est que l'utilisateur peut décider

d'effacer certains clusters. Dans ce cas, les visualisations contenant ce cluster devront se mettre à jour automatiquement.

Durant la dixième semaine de stage, j'ai commencé le développement des plugins d'exportation. L'ancien développeur avait laissé des travaux pour simuler une gestion de plugins. Après un point avec M. DEMKO, j'ai donc modifié les travaux existants afin de rendre l'exportation utilisable sous forme de plugin. J'ai créé un nouveau projet Poetry pour développer la méthode d'exportation. Même si les plugins seront installés indépendamment du projet principal, ils devront suivre une certaine logique pour fonctionner correctement. Par exemple, les plugins devront posséder le même chemin d'accès pour être reconnus. Ils doivent également utiliser le même nom de méthode pour que l'application principale puisse les utiliser. Pour ce faire, une classe abstraite a été ajoutée dans le code du projet principal ; tous les plugins d'exportation devront hériter de cette classe abstraite pour fonctionner. Les différents types de plugins (comme les visualisations) devront suivre le même schéma pour fonctionner. Une fois les plugins d'exportation développés et installés, le projet principal recherchera les paquets Python installés selon un certain nom (chemin d'accès) et chargera chaque plugin d'exportation. L'utilisateur aura alors un menu déroulant avec toutes les extensions disponibles pour l'exportation de son projet.

M. DEMKO a également évoqué différents axes d'amélioration pour ces plugins, car la première version nécessite d'installer un à un les plugins d'exportation, ce qui peut vite devenir long et redondant. La première solution proposée est de rassembler certains types d'extensions dans un seul plugin. L'autre problème concerne l'exportation de certaines extensions. Si un utilisateur souhaite exporter un fichier ou un projet en format zip, la marche à suivre ne sera pas forcément la même. Il sera donc nécessaire d'ajouter une autre fenêtre "pop-up" pour laisser l'utilisateur choisir certains paramètres, et d'enregistrer ces paramètres pour éviter à l'utilisateur de devoir les saisir à nouveau à chaque utilisation.

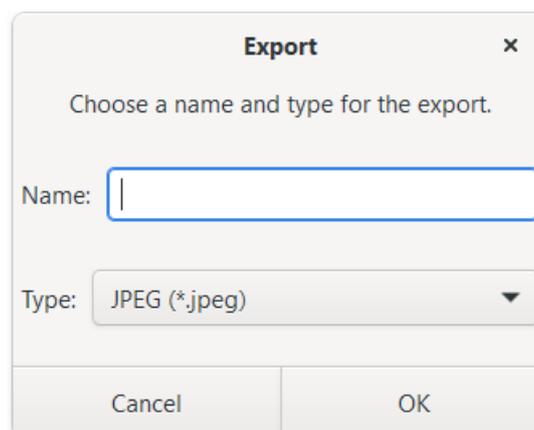


Figure 18: Fenêtre pop-up pour l'exportation

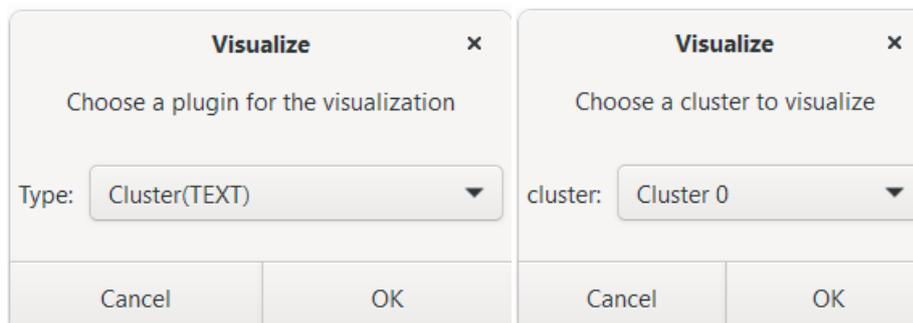
J'ai ensuite enchaîné sur le développement des plugins de type visualisation. Pour gérer la première contrainte de ce plugin, qui est la gestion soit d'un cluster unique soit d'un groupe de clusters, j'ai fait apparaître une fenêtre pop-up dans laquelle l'utilisateur peut choisir s'il veut visualiser un cluster unique ou un groupe entier. Cette fenêtre est suivie par une seconde pop-up où l'utilisateur peut choisir un cluster unique s'il a sélectionné la première option ou un groupe de clusters dans l'autre

cas. Il pourra également choisir le plugin qu'il veut utiliser pour obtenir la visualisation souhaitée.

Pour le moment, j'ai développé un seul type de visualisation, basé sur une méthode de l'ancien développeur, qui affiche les détails d'un cluster. Une fois la manipulation effectuée par l'utilisateur, une ligne s'ajoute dans le menu "Visualisations" de l'interface. Si l'utilisateur interagit avec cette ligne par un double-clic, une fenêtre va apparaître avec les détails du cluster choisi, ou une série de fenêtres pour chaque cluster d'un groupe.

Finalement, ce type de plugin contient lui-même deux sous-types de plugins, et la marche à suivre ne sera pas forcément la même selon qu'on choisit de visualiser un cluster unique ou un groupe de clusters. Pour ce type de plugin, j'ai également dû gérer les visualisations créées avec les clusters et les groupes présents dans un projet. En effet, si un utilisateur souhaite supprimer un élément dans son projet, cela aura sans doute un impact sur les visualisations générées.

Pour empêcher le moindre problème, j'ai ajouté une vérification dans les plugins afin qu'ils soient alertés lorsqu'un élément du projet (cluster ou groupe) est supprimé. Cette vérification va ensuite supprimer la ligne dans le menu de visualisation dans le cas d'un cluster. Pour les groupes, le mécanisme sera différent, car même si un cluster est supprimé, le groupe sera toujours présent. La visualisation prendra donc en compte l'état actuel du groupe et affichera tous les clusters qui y sont présents. Si le groupe entier est supprimé, la ligne correspondante dans le menu "Visualisations" sera également supprimée.



3.5 Ajout fenêtres secondaires

3.5.1 Fenêtre d'introduction

Afin d'obtenir une application ergonomique et agréable à utiliser, l'ergonome (M. GAUTREAU) a eu l'idée d'ajouter une fenêtre d'introduction à GALACTIC-Explorer lors de son démarrage. Au lieu de ne rien afficher en attendant le chargement de l'application, l'utilisateur se verra présenter une fenêtre contenant une illustration du portrait d'Évariste Galois réalisée par M. GAUTREAU lui-même, ainsi que le logo, le nom de l'application et une brève description. Les couleurs de cette fenêtre ont déjà été discutées avec l'ergonome pour obtenir un rendu satisfaisant. Cette fenêtre apparaîtra donc dès que l'utilisateur ouvrira GALACTIC-Explorer et restera affichée pendant 5 secondes avant de faire place à la fenêtre d'accueil.

Pour obtenir un résultat aussi fidèle que possible au croquis de M. GAUTREAU, j'ai dû positionner manuellement les différents éléments de la page de présentation en utilisant des pixels. Bien que

la bibliothèque GTK permette de positionner les éléments graphiques de manière personnalisée, la disposition reste sommaire : un élément prendra soit la taille minimale possible, soit la taille maximale, ce qui rend la disposition de la fenêtre compliquée sans avoir recours aux pixels.



Figure 19: Page de présentation GALACTIC-Explorer

3.5.2 Fenêtre d'accueil

Cette fenêtre se divise en plusieurs parties et permet à un utilisateur novice d'être guidé pas à pas pour utiliser GALACTIC-Explorer. Les premiers éléments visibles de cette fenêtre sont le logo de l'application ainsi qu'une barre de menu.

- Le menu DASHBOARD correspond à la fenêtre d'accueil principale où l'utilisateur peut créer un nouveau projet ou en ouvrir un autre déjà existant.
- Le menu GALACTIC CONFIGURATOR permet à l'utilisateur de créer des fichiers de type data ou de type explorer s'il n'en a pas encore à disposition. Pour l'instant, ce menu est seulement décoratif car la fonctionnalité GALACTIC CONFIGURATOR n'est pas encore fonctionnelle.
- Le menu GALACTIC MARKET permet à l'utilisateur d'ajouter différents types de plugins à son application principale pour ajouter des fonctionnalités réalisées soit par un développeur lié à GALACTIC soit par un utilisateur de la communauté. Ce menu est également décoratif car les modalités d'installation de plugins sont encore en discussion.

La partie DASHBOARD de la fenêtre d'accueil est la partie principale que j'ai réalisée. Après la barre de menu, l'utilisateur peut créer un nouveau projet. Il commence par entrer un nom pour le projet ainsi qu'une courte description. Ensuite, 3 boutons le dirigent vers l'arborescence de son système pour qu'il détermine l'emplacement du projet ainsi qu'un chemin d'accès vers un fichier de type data et de type explorer. Cette première étape est assez guidée et l'utilisateur doit obligatoirement remplir tous ces

critères pour créer son projet. Dans le cas contraire, l'application affiche des fenêtres pop-up pour chaque élément oublié afin d'éviter tout problème de fonctionnement.

L'autre partie de la fenêtre d'accueil permet à l'utilisateur de chercher un projet existant dans l'arborescence de son système et d'ouvrir l'un des 4 projets qu'il a créés ou ouverts récemment. J'ai enregistré le nom des projets et leur chemin d'accès dans les préférences utilisateur. À chaque nouvelle utilisation de l'application, les préférences utilisateur affichent les 4 derniers projets utilisés par l'utilisateur pour lui faire gagner du temps.

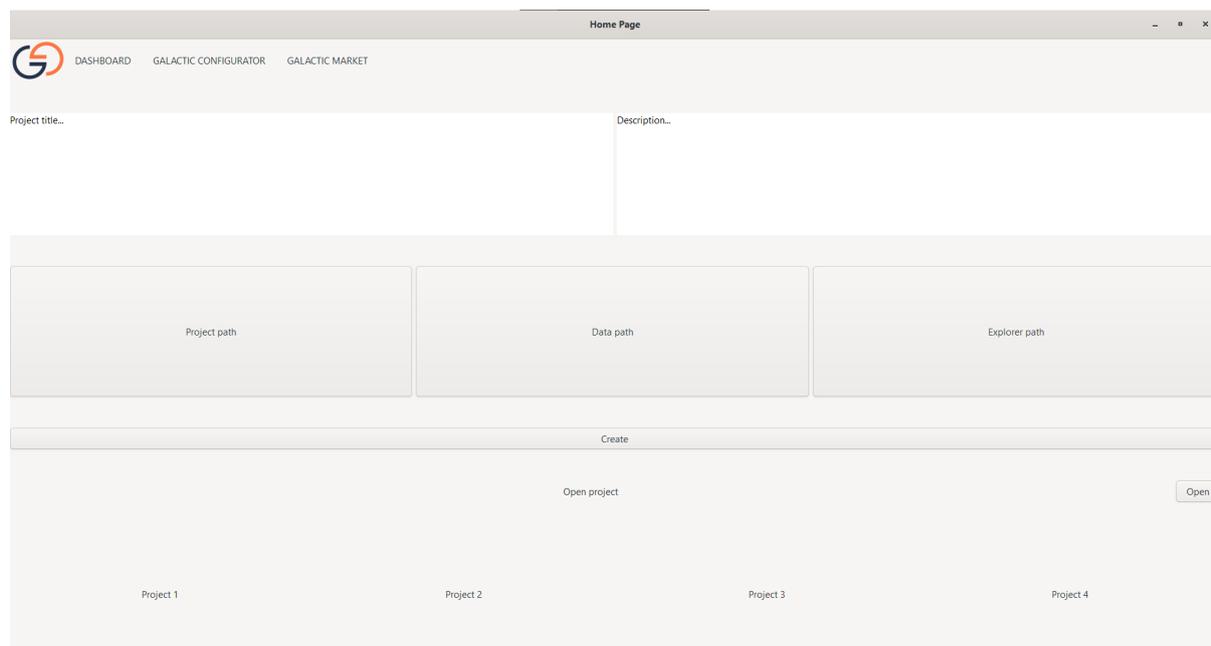


Figure 20: Page d'accueil GALACTIC-Explorer

3.6 Préférences utilisateur

L'un des points importants pour améliorer l'utilisation de l'interface GALACTIC-Explorer est la mise en place des préférences utilisateur. Cette fonctionnalité permet d'enregistrer certains paramètres afin qu'ils soient chargés à chaque nouvelle ouverture de l'application. Pour mettre en place cette fonctionnalité, j'ai utilisé la bibliothèque Python `appdirs`, qui permet d'enregistrer n'importe quel type de fichier dans un dossier spécifique. La particularité de cette bibliothèque est que le chemin d'accès vers ce dossier est universel et n'est pas directement rentré en dur dans le code, ce qui signifie qu'un utilisateur pourra ensuite utiliser GALACTIC-Explorer sur n'importe quel système d'exploitation. Une fois cette bibliothèque installée, il suffit d'y ajouter un fichier pour enregistrer différents paramètres. M.DEMKO m'a conseillé d'utiliser des fichiers de type YAML afin de permettre aux utilisateurs de modifier ces fichiers à la main si nécessaire.

Les premiers paramètres que j'ai enregistrés dans ce fichier sont des variables de type booléen (Vrai/Faux) afin de garder en mémoire le poste de travail de l'utilisateur. S'il souhaite garder les menus à gauche et à droite de l'application cachés ou même détachés de l'interface principale, le code enregistre simplement ces paramètres lors de la fermeture de l'application, et lorsque l'utilisateur revient dessus, le fichier est lu pour repositionner les menus comme ils étaient lors de la dernière session.

Avec les préférences utilisateur, j'ai également ajouté une fenêtre de paramètres permettant à l'utilisateur d'ajouter des chemins d'accès par défaut pour différents dossiers :

- Dossier où se situent tous les projets de GALACTIC-EXPLORER
- Dossier où se situent tous les fichiers de type data
- Dossier où se situent tous les fichiers de type explorer

Ainsi, lorsque l'utilisateur souhaite ouvrir un projet ou ajouter un fichier de type data ou de type explorer dans un nouveau projet, il n'a pas besoin de les chercher à partir de la racine de l'arborescence de son système, ce qui permet un gain de temps.

Pour faire gagner du temps aux utilisateurs aguerris de GALACTIC-Explorer, j'ai également ajouté deux boutons de type "check" dans la fenêtre des paramètres pour autoriser ou non l'affichage des deux autres fenêtres présentes au début du lancement du logiciel. La première fenêtre d'introduction à GALACTIC-Explorer reste affichée pendant 5 secondes après le démarrage, tandis que la fenêtre d'accueil guide les utilisateurs novices pour créer leur propre projet ou ouvrir leurs autres projets déjà créés. L'utilisateur peut donc choisir de les afficher ou non directement dans les paramètres pour également gagner du temps.

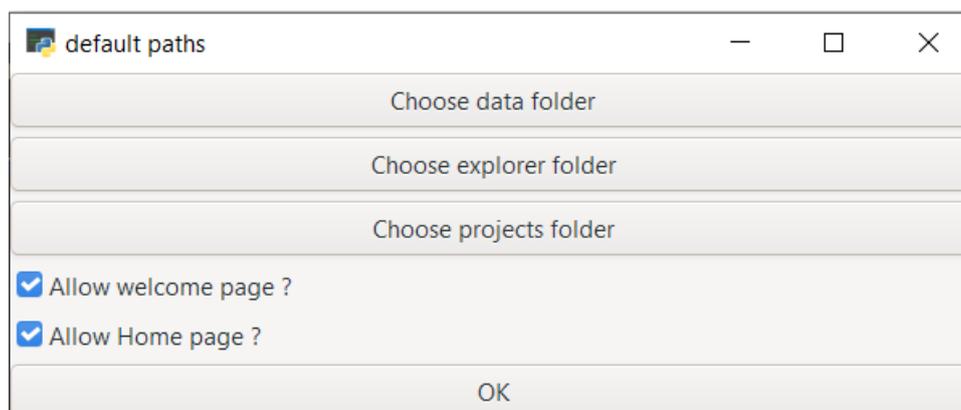


Figure 21: Fenêtre préférence utilisateur

3.7 Gestion de l'application

Lors de la prise en main de l'application au début de mon stage, l'utilisateur ne pouvait enregistrer son travail qu'à l'aide d'un simple bouton. Avec les différentes modifications que j'ai apportées, j'ai donné le choix à l'utilisateur de pouvoir enregistrer son projet dans de nombreux cas afin de garantir qu'il ne perde pas sa progression en raison d'une mauvaise manipulation.

Le premier cas de sauvegarde que j'ai ajouté est la sauvegarde avant de quitter l'application. Si l'utilisateur décide de fermer l'application alors que son projet est ouvert, l'interface lui demandera s'il souhaite oui ou non enregistrer son travail, ou même annuler cette action s'il a fait une fausse manipulation.

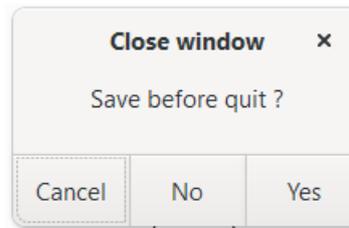


Figure 22: Fenêtre de sauvegarde avant fermeture

Le deuxième cas de sauvegarde est la sauvegarde avant d'ouvrir un autre projet dans la même fenêtre. Si l'utilisateur souhaite ouvrir un projet alors qu'un autre est déjà ouvert, l'interface lui proposera une fenêtre pop-up, comme pour le cas de la sauvegarde avant de quitter.

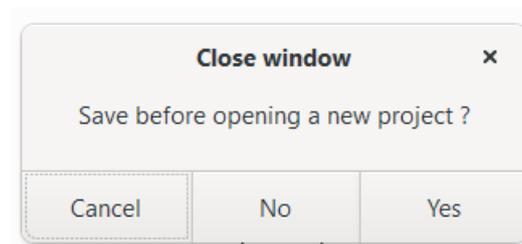


Figure 23: Fenêtre de sauvegarde avant changement de fenêtre

L'utilisateur peut également gérer ses projets en ouvrant plusieurs fenêtres à la fois, ce qui lui permet de travailler sur plusieurs projets simultanément sans avoir à les ouvrir un à un dans la même fenêtre.

3.8 Nettoyage + restructuration du code

Après les modifications apportées à l'interface, j'ai fait une pause dans le développement pour nettoyer mon code. L'ancien développeur avait laissé plusieurs outils qui fonctionnaient comme des tests et qui suivaient des règles de codage très précises. Naturellement, la première fois que j'ai lancé ces tests, ils m'ont renvoyé une liste d'erreurs que j'ai dû corriger au fur et à mesure. Plusieurs règles doivent être respectées pour avoir un code conforme :

- Éviter les espaces inutiles en fin de lignes.
- Ne pas avoir plus d'un saut de ligne entre les lignes de code.
- Les commentaires de type docstring (au début de chaque méthode) doivent contenir une seule ligne de texte se terminant par un point, un espace, puis une description.
- Placer les importations de paquets dans un ordre précis.
- Préciser le type des variables dans les méthodes.
- Ne pas dépasser un certain nombre de caractères sur une ligne pour faciliter sa lecture.
- La première ligne d'un docstring doit être à l'impératif.

4 Travail en cours + prévision

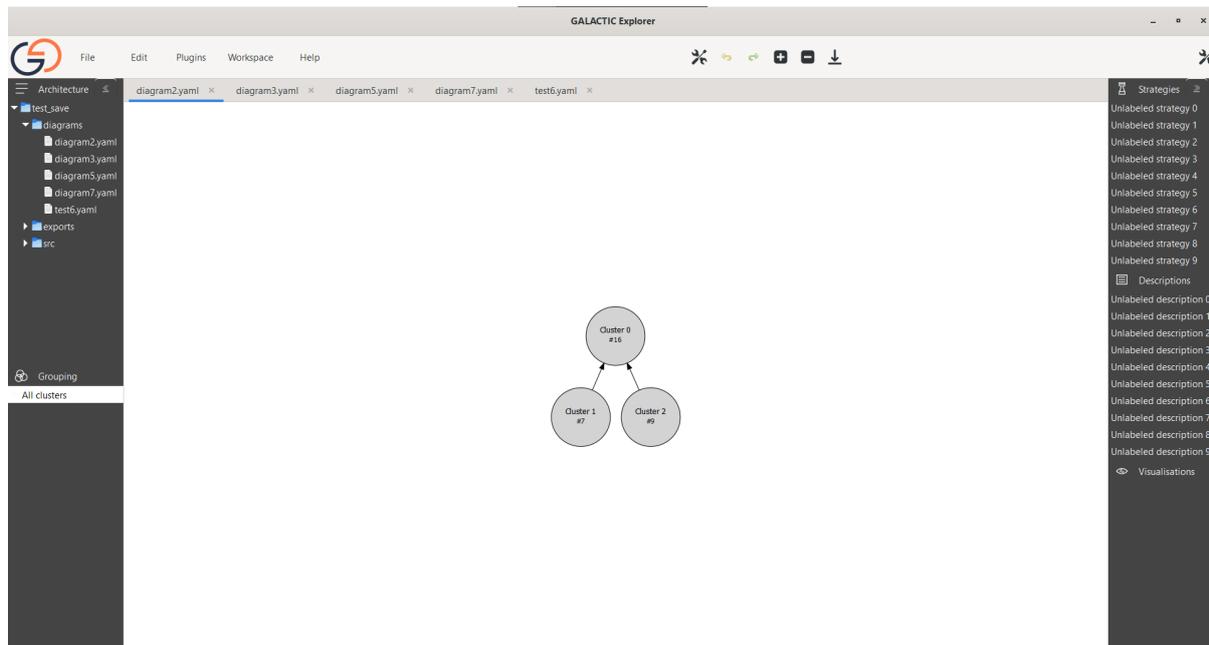


Figure 24: Aperçu interface principale

Au moment de la rédaction de ce rapport, ma période de stage n'est pas encore terminée. Si je gère bien mon temps pour achever ce rapport ainsi que les diapositives de présentation, il me restera environ trois semaines de travail pour poursuivre l'amélioration, voire finaliser, l'interface de GALACTIC-Explorer. L'objectif principal fixé avec M. DEMKO et l'ergonome M. GAUTREAU était de réaliser une première version fonctionnelle de l'application, suite à la réunion du 8 mars. Lors de cette réunion, plusieurs points ont été abordés et un croquis m'a été fourni par M. GAUTREAU pour avoir une idée précise du produit final. Avec le travail accompli depuis le début de mon stage, je peux affirmer que la première version demandée par M. DEMKO et M. GAUTREAU est achevée. Cependant, certains aspects restent à améliorer, sur lesquels je vais devoir travailler au cours des trois dernières semaines de stage afin de finaliser le projet.

La plupart des points restants sont des tâches laissées par l'ancien développeur, M. DESCHAMPS, après discussion avec M. DEMKO. J'avais commencé à travailler sur ces tâches, mais je les ai laissées de côté après la réunion du 8 mars, car de nouvelles directives étaient données et il me semblait plus important de livrer la première version dans les temps plutôt que d'ajouter des fonctionnalités que je considérais comme secondaires.

L'une des tâches que je souhaite réaliser avant de terminer mon stage est la traduction automatique de l'interface. J'ai déjà commencé à travailler dessus en recherchant une bibliothèque Python pouvant fournir de tels outils, et j'ai réussi à en trouver une qui propose plusieurs moyens de traduction automatique, tels que Google Traduction ou même ChatGPT. Bien que les traductions ne soient pas fiables à 100 %, elles permettent tout de même de comprendre le sens du texte. Cependant, j'ai rencontré plusieurs problèmes. Tout d'abord, le temps d'exécution était trop long, car l'interface contient de nombreuses chaînes de caractères réparties dans différentes fenêtres. Ainsi, à chaque

lancement de l'application, il fallait attendre au moins 30 secondes avant de pouvoir l'utiliser, et il y avait également des temps de chargement lors de l'apparition de nouvelles fenêtres. Bien que la bibliothèque soit fonctionnelle, ces temps de chargement nuisent à l'ergonomie de l'application, qui est l'objectif principal de mon stage. De plus, M. DEMKO ne souhaitait pas utiliser cette bibliothèque, car il préférerait un système de traduction natif, sans recourir à une bibliothèque tierce. Pour ne pas perdre trop de temps sur la mise en place de cette fonctionnalité, j'ai décidé de la laisser de côté pour le moment. Après une discussion avec M. DEMKO, nous avons convenu que cette fonctionnalité serait mise en attente, car la contrainte d'avoir un système de traduction natif est complexe. M. DEMKO a déjà une idée sur la manière de procéder, mais il souhaite l'améliorer avant de l'implémenter dans GALACTIC-Explorer. Je vais donc me concentrer sur d'autres tâches et mettre toutes les chaînes de caractères de l'application soit en français, soit en anglais.

Certaines tâches sont également en suspens pour le moment, et je travaillerai dessus si le temps me le permet :

- Ajouter une fonctionnalité de retour arrière en cas de suppression d'un fichier, afin que l'utilisateur puisse le récupérer en cas de fausse manipulation.
- Permettre l'ouverture des visualisations (obtenues grâce au plugin homonyme) en tant qu'image dans la même zone que celle des diagrammes de Hasses générés par l'application.
- Ajouter ces visualisations de type images dans l'arborescence du projet.
- Ajouter un bouton "supprimer" sur chaque ligne du menu "grouping". J'ai déjà essayé de réaliser cette tâche, mais GTK ne permet pas d'ajouter des objets de type bouton dans un menu de type treeview. Bien qu'une certaine manipulation permette probablement d'obtenir cette fonctionnalité, je n'ai pas réussi à le faire. Comme la suppression d'un certain groupe est possible à l'aide d'un clic droit, je n'ai pas voulu perdre trop de temps sur cette tâche.
- Ajouter des raccourcis clavier pour obtenir les descendants ou les prédécesseurs d'un certain cluster sur le diagramme de Hasses.
- Ajouter une barre de chargement pour les projets avec des temps d'exécution lents.
- Ajouter dans le fichier de sauvegarde du diagramme (.yaml) les dernières actions de l'utilisateur, afin qu'il puisse revenir en arrière même après la fermeture de l'interface.

Un autre point abordé lors de la réunion est la mise en place d'un thème GTK afin d'utiliser la charte graphique de M. GAUTREAU directement sur l'application. Sachant qu'avec GTK, il est possible de créer son propre thème, un utilisateur pourra même, s'il le souhaite, utiliser un autre thème si celui par défaut de l'application ne lui convient pas. Les dernières versions de GTK3 utilisent le CSS pour la mise en page, donc cela ne devrait pas poser de véritable difficulté pour moi.

Un autre élément discuté lors de la réunion, mais qui n'a pas encore abouti, est la gestion des plugins installés sur l'application. Deux solutions possibles ont été évoquées :

- Mise en place d'une fenêtre de gestion des plugins permettant de les supprimer facilement, sans avoir besoin de le faire manuellement dans l'arborescence. L'utilisateur n'aurait qu'à cocher les plugins qu'il souhaite supprimer, et l'interface s'en chargerait automatiquement. Cependant, l'installation resterait manuelle, ce qui pourrait rapidement devenir fastidieux.
- Mise en place d'un "Market" permettant de gérer à la fois la suppression et l'installation des plugins, avec une barre de recherche pour trouver plus facilement les plugins souhaités. Cette

solution offrirait un portail en ligne servant de hub pour partager les plugins avec la communauté. Elle pourrait également permettre la vente de certains plugins réalisés par les développeurs liés au projet GALACTIC-Explorer, dans le cadre de la commercialisation de cette application.

Après la rédaction de ce rapport et la préparation de ma soutenance, je ferai probablement le point avec M. DEMKO pour déterminer quelles tâches seraient les plus intéressantes à réaliser pour améliorer l'interface.

5 Difficultés rencontrés

J'ai réalisé une auto-formation sur la librairie GTK en Python. La principale difficulté rencontrée a été la documentation de GTK. L'ancien développeur avait laissé un lien vers une documentation pour GTK en Python qui m'a été très utile, car la plupart des ressources trouvées sur Internet concernent la version C de GTK.

Une autre difficulté a été de m'approprier un code réalisé par une autre personne. Celui-ci étant conséquent, il m'a fallu un certain temps avant de le comprendre totalement. En effet, rien que le fichier principal contient plus de 3 000 lignes. J'ai donc pris le temps de lire et de comprendre chaque méthode développée par l'ancien développeur afin d'éviter de développer des fonctionnalités qui existent déjà. Par exemple, le développeur avait ajouté des méthodes permettant d'ajouter un sous-menu à un menu déjà existant, ou de créer directement une fenêtre pop-up avec seulement quelques paramètres à donner. Bien que cette compréhension m'ait pris un certain temps, elle m'a également été bénéfique, car j'ai pu utiliser les méthodes du développeur pour améliorer l'interface.

Le projet en lui-même est assez complexe. Rien que l'utilisation de GTK est complexe, car cette bibliothèque possède un grand nombre de classes différentes pour ajouter diverses fonctionnalités à son interface. Ces mêmes classes possèdent également un nombre important de méthodes pour avoir des fonctionnalités soit génériques soit personnalisées. La mise en forme de l'interface a également été complexe, car GTK fonctionne comme des poupées russes. Une interface est un bloc principal qui va contenir différents sous-blocs, qui vont eux-mêmes contenir d'autres sous-blocs. Ce fonctionnement permet de disposer les éléments de notre interface comme on le souhaite.

Avec la multitude de tâches qui m'ont été confiées, j'ai dû gérer mon temps de travail pour ne pas passer trop de temps sur telle ou telle tâche. Aucune tâche n'était liée entre elles, et je devais donc les réaliser une par une plutôt que de travailler sur plusieurs tâches en même temps. Cela m'a permis de réaliser des commits (sauvegardes) sur le dépôt Git de GALACTIC-Explorer de l'interface fonctionnelle plutôt que de faire des commits de tâches non complétées, ce qui aurait pu rendre le code non fonctionnel.

6 Activités au sein de l'entreprise

Durant ma période de stage, j'ai été convié à un événement du Laboratoire Informatique, Images et Interaction de La Rochelle. Cet événement, nommé Journées Scientifiques, s'est déroulé les jeudi 16 et vendredi 17 mai et est organisé chaque année par le laboratoire afin de présenter brièvement les

différents travaux au sein des différentes équipes du laboratoire. Durant ces deux journées, j'ai donc pu assister tout d'abord à une Assemblée Générale du laboratoire afin de comprendre le fonctionnement du laboratoire du point de vue budgétaire, mais également sur d'autres points tels que les départs et les arrivées des personnes travaillant au sein du laboratoire. Ensuite, chaque équipe du laboratoire (Modèles et Connaissances, Images et Contenus, Dynamique des Systèmes et Adaptabilité) a présenté leurs nouveaux projets ainsi que l'avancement des projets en cours. Il m'a été demandé de présenter l'avancement de mes travaux durant la présentation de mon équipe, Modèles et Connaissances. Comme pour le rapport intermédiaire, cette petite présentation m'a permis de faire un point sur mon avancement, mais également de réaliser un premier entraînement pour ma soutenance le 24 juin 2024. J'ai également pu voir les différents projets de mon équipe ainsi que ceux des autres équipes. Tous ont été intéressants et variés, et m'ont également permis d'avoir une vision globale du laboratoire dans lequel j'effectue mon stage.

J'ai eu l'occasion de participer à un séminaire qui s'est tenu le jeudi 23 mai. Ce séminaire avait pour sujet "Trajectoires de Territoire et Trajectoires de Vie : Vers d'Autres Types de Trajectoires Sémantiques" et a été présenté par Jérôme Gensel, professeur à l'Université Grenoble Alpes. Durant ce séminaire, M. Gensel nous a d'abord parlé plus en détail des trajectoires, mais également des trajectoires sémantiques avec les différents modèles existants. Ensuite, M. Gensel nous a parlé des trajectoires de vie, en présentant notamment un modèle spécifique (SALTO) pour les parcours biographiques. Enfin, il a conclu en nous présentant les trajectoires de territoires en nous montrant le modèle TRACES sur les trajectoires environnementales du territoire.

7 Conclusion

Lors de mon stage de fin d'études en Master informatique (spécialité Architecte logiciel), qui s'est déroulé du 05 février au 21 juin 2024, j'ai eu le privilège de travailler avec M. DEMKO sur son projet : GALACTIC-Explorer : une interface interactive pour galactic.

L'objectif principal de ce projet était d'apporter une amélioration globale à une interface graphique déjà réalisée par le développeur Etienne DESCHAMPS. Cela comprenait l'ajout de nouvelles fonctionnalités pour faciliter l'utilisation de l'interface pour un utilisateur novice, ainsi que l'intégration de différents raccourcis pour faciliter la vie des utilisateurs expérimentés. Il s'agissait également d'améliorer l'ergonomie globale de l'application pour la rendre aussi conviviale que possible. Une autre composante essentielle était l'implémentation d'un système de plugins pour certaines fonctionnalités, comme l'exportation de diagrammes ou la visualisation des clusters dans ce même diagramme.

Tous ces aspects ont été abordés lors d'une réunion qui a eu lieu un mois après le début de mon stage. Pendant cette période, j'ai pu m'approprier les travaux déjà réalisés par M. DESCHAMPS et comprendre le fonctionnement de GTK, une bibliothèque Python sur laquelle repose l'intégralité de l'interface. Cette réunion a permis d'orienter mon travail en me donnant des objectifs clairs. Grâce à la collaboration avec l'ergonome M. GAUTREAU, j'ai obtenu un fil directeur précis pour mon stage, avec différents croquis pour m'orienter. Il ne me restait plus qu'à utiliser mes connaissances théoriques et pratiques pour réaliser une première version fonctionnelle de cette interface.

Malgré certaines difficultés rencontrées, telles que l'apprentissage d'une bibliothèque complexe (GTK)

et le fait de travailler à partir d'une base réalisée par un autre développeur, il reste encore quelques tâches à accomplir que je vais tenter de réaliser dans le temps qu'il me reste afin de proposer une version finale la plus complète possible. Même si l'objectif principal du projet, tel que vu par M. DEMKO et M. GAUTREAU, est atteint du point de vue de la première version fonctionnelle.

Cependant, même si cette première version est fonctionnelle, il reste encore plusieurs axes d'amélioration pour la rendre commercialisable. Par exemple, la gestion des plugins via un "Marketplace" pour rechercher plus facilement des plugins réalisés par des développeurs liés au projet GALACTIC ou par la communauté. Un autre aspect important à envisager serait l'installation de GALACTIC-Explorer sur différents systèmes d'exploitation existants. En effet, durant mon stage, j'ai dû utiliser l'outil MSYS2 pour travailler sous Windows, car la bibliothèque GTK ne fonctionnait pas correctement autrement.

8 Conclusion (english)

During my final internship for my Master's degree in Computer Science (specializing in Software Architecture), which took place from February 5th to June 21st, 2024, I had the privilege of working with Mr. DEMKO on his project: GALACTIC-Explorer: an interactive interface for Galactic.

The main objective of this project was to provide overall improvement to a graphical interface already developed by the developer Etienne DESCHAMPS. This included adding new features to make the interface easier to use for novice users, as well as integrating various shortcuts to make life easier for experienced users. It also aimed to improve the overall ergonomics of the application to make it as user-friendly as possible. Another essential component was the implementation of a plugin system for certain functionalities, such as exporting diagrams or visualizing clusters within the same diagram.

All these aspects were discussed during a meeting held one month after the start of my internship. During this period, I was able to familiarize myself with the work already done by Mr. DESCHAMPS and understand how GTK, a Python library on which the entire interface is based, works. This meeting guided my work by providing clear objectives. Thanks to collaboration with the ergonomist Mr. GAUTREAU, I obtained a precise guideline for my internship, with various sketches to guide me. All that was left was to use my theoretical and practical knowledge to develop a first functional version of this interface.

Despite some difficulties encountered, such as learning a complex library (GTK) and working from a codebase developed by another developer, there are still some tasks to be completed, which I will attempt to accomplish within the remaining time to offer the most complete final version possible. Even though the main objective of the project, as seen by Mr. DEMKO and Mr. GAUTREAU, has been achieved from the perspective of the first functional version.

However, even though this first version is functional, there are still several areas for improvement to make it marketable. For example, managing plugins via a "Marketplace" to easily search for plugins developed by developers associated with the GALACTIC project or by the community. Another important aspect to consider would be installing GALACTIC-Explorer on different existing operating systems. Indeed, during my internship, I had to use the MSYS2 tool to work on Windows, as the GTK library did not function correctly otherwise.

9 Apport du stage/Projet futur

9.1 Apport du stage

Pour la deuxième fois consécutive, j'ai eu la chance de travailler sur un sujet de stage proposé par M. DEMKO, tout comme lors de mon premier stage réalisé lors de ma première année de Master, portant sur le développement d'un plugin en PHP pour la plateforme Moodle. Ce nouveau stage a été bénéfique pour moi à plusieurs égards.

Tout d'abord, sa durée a été un avantage majeur, me permettant de m'investir pleinement dans ce projet pendant mes 20 semaines de stage, en prenant le temps nécessaire pour réaliser les différentes tâches qui m'ont été confiées. Bien que M. DEMKO ait suivi de près l'avancement de mon projet, j'ai bénéficié d'une certaine autonomie et d'une gestion du temps qui m'ont obligé à être régulier dans mon travail tout au long du stage.

Ce stage m'a permis d'améliorer mes compétences en langage Python et m'a également initié à la bibliothèque d'interface GTK. J'ai réalisé une auto-formation sur cette librairie en parallèle de mon avancée sur le projet GALACTIC-Explorer. De plus, j'ai acquis une compétence précieuse pour le monde professionnel : travailler sur une base de code réalisée par un tiers. Au cours de mes études universitaires, la plupart des projets que j'ai réalisés partaient de zéro. Pour la première fois, j'ai dû m'approprier un code qui n'était pas le mien, prenant le temps de le comprendre, étant donné qu'il était assez conséquent (+3 000 lignes de code), afin de ne pas développer de fonctionnalités déjà présentes.

J'ai également eu la chance de collaborer avec un professionnel, l'ergonome M. GAUTREAU, qui m'a guidé sur la manière de rendre une interface ergonomique et agréable à utiliser.

9.2 Projet futur

Une fois mon master terminé, je vais entreprendre une recherche intensive pour décrocher mon premier emploi et entrer véritablement dans le monde du travail. Fort des compétences acquises tout au long de mon parcours universitaire et de mes divers stages au sein du laboratoire Informatique, Images et Interaction, j'espère pouvoir trouver un poste de développeur pour perfectionner mes compétences.

Bien que les stages effectués au sein du laboratoire Informatique, Images et Interaction de La Rochelle ne reflètent pas entièrement le monde professionnel, ils m'ont permis de mettre en pratique les compétences acquises lors de mes études universitaires. De plus, ces stages m'ont offert l'opportunité d'acquérir de nouvelles compétences essentielles pour le monde du travail, telles que le travail en collaboration, l'autonomie et la gestion du temps pour mener à bien un projet dans les délais impartis, ainsi que l'auto-formation sur des technologies ou des bibliothèques complexes.

10 Sitographie

Documentation GTK (Python) (05/02/2024) [<http://lazka.github.io/pgi-docs/Gtk-3.0/index.html>]

Documentation Poetry (05/02/2024) [<https://python-poetry.org/docs/>]

Documentation GALACTIC (05/02/2024) [<https://galactic.univ-lr.fr/>]

Page bibliothèque appdirs (06/05/2024) [<https://pypi.org/project/appdirs/>]

Page bibliothèque deeptranslator (13/05/2024) [<https://pypi.org/project/deep-translator/>]

Documentation GALACTIC (05/02/2024) [<https://galactic.univ-lr.fr/>]