



INRAE (UMR SAS, RENNES)
UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE
(GÉNIE INFORMATIQUE)

RAPPORT DE STAGE DE FIN D'ÉTUDES (TN10)

**Développement d'un module
MEANS-InOut de calcul d'impacts
environnementaux de systèmes agricoles**

Clément BRIZARD

supervisé par
Caroline MALNOË (INRAE)
Jean-Benoist LÉGER (UTC)

février - juillet 2020

Remerciements

À l'UMR SAS, pour son accueil et son animation. Merci en particulier aux autres stagiaires et doctorants.

À Caroline Malnoë, pour son accompagnement précieux tout au long du stage.

À Julie Auberger, pour son aide et son expertise agronomique.

À Julien Hilaire et Guilhem Rostain, pour leur aide tout au long du stage.

À Joël Aubin, pour avoir rendu possible et valorisé ce stage.

À Jean-Benoist Léger, pour son aide et son accompagnement, avant et pendant le stage.

Résumé

Le stage présenté dans ce rapport visait à développer un module de calcul d'impacts environnementaux pour MEANS-InOut, logiciel pour l'évaluation des systèmes agricoles développé par l'UMR SAS à INRAE de Rennes, et le Cirad à Montpellier. L'outil s'appuie sur la méthode de l'Analyse du Cycle de Vie (ACV) et la méthode AGRIBALYSE développée par l'ADEME. Initialement, les calculs d'impacts devaient être réalisés avec un logiciel tiers, en l'occurrence SimaPro, référence des logiciels d'ACV mais propriétaire et sans API. La première partie du stage a permis de choisir un outil d'ACV open-source et facilement intégrable avec MEANS-InOut : openLCA. Le reste du stage a été consacré à l'intégration des deux outils via le format XML ecoSpold1 et au développement du module de calcul. Une partie non-négligeable du travail a dû être dédiée à l'alignement des bases de données d'ACV avec openLCA, celles de MEANS-InOut ayant été formatées par les exigences de SimaPro. Ce rapport présente le cadre opérationnel et méthodologique du stage, avant de détailler ses principales réalisations et les choix effectués au cours du travail.

Abstract

The six-month internship presented in this report aimed to develop an environmental impact calculation module for MEANS-InOut, software for the evaluation of agricultural systems developed by UMR SAS at INRAE in Rennes and CIRAD in Montpellier. The tool is based on the Life Cycle Assessment (LCA) method and the AGRIBALYSE method developed by ADEME. Initially, the impact calculations were carried out with a third-party software, in this case SimaPro, the reference for LCA software, but closed-source and without API. The first month of the training course was focused on choosing an open-source LCA tool that could be easily integrated with MEANS-InOut : openLCA. The rest of the internship was devoted to the integration of the two tools via the XML ecoSpold1 format and to the development of the calculation module. A non-negligible part of the work had to be dedicated to the alignment of the LCA databases with openLCA, those of MEANS-InOut having been formatted by SimaPro requirements. This report presents the operational and methodological framework of the internship, before detailing its main achievements and the choices made during the work.

Sommaire

I	Présentation de l'UMR SAS et de MEANS	5
1	UMR SAS	6
1.1	Définition et composition	6
1.2	Domaine de recherche	6
1.3	Organisation	7
2	MEANS	7
2.1	Définition	7
2.2	Gouvernance	8
2.3	Fonctionnement de l'équipe	9
2.4	MEANS-InOut	9
2.5	L'ACV et MEANS-InOut	9
II	Mission	11
3	Présentation du sujet	12
3.1	Motivations	12
3.2	Sujet initial et évolutions	14
4	Contributions	15
4.1	Outil de mapping entre MEANS-InOut et openLCA	15
4.2	Outil de calcul d'impacts via openLCA	16
5	Planning	16
5.1	Déroulement effectif du stage	16
5.2	Étapes principales	17
6	Outils et technologies	18
6.1	Développement	18
6.2	Organisation du travail	18
6.3	Tests	19
7	Prise de recul	19
III	Réalisations	20
8	Lancement du stage	21
8.1	Choix de l'outil de calcul	21

SOMMAIRE

8.2	Choix du format d'échange de données	23
9	Développement	25
9.1	Préparation de l'import des objets ecoSpold1	25
9.2	Mapping entre MEANS-InOut et openLCA	28
9.3	Implémentation du calcul d'impacts	33
9.4	Améliorations	35
9.5	Comparaison des résultats avec SimaPro	36
9.6	Extension aux successions de cultures	38
	Conclusion	39
	Bibliographie	40
	Acronymes	41
	Glossaire	42

Table des figures

2.1	Organigramme des différentes instances de la plate-forme MEANS	8
2.2	Fonctionnement d'une analyse environnementale avec MEANS-InOut	10
3.1	Objectif principal du stage	13
9.1	Ajout de l'import dans openLCA au traitement d'un ITK par MEANS-InOut	26
9.2	Fonctionnalités d'un driver embarqué Derby	27
9.3	Nécessité du mapping entre MEANS-InOut et openLCA	28
9.4	UML simplifié du modèle de données d'openLCA	29
9.5	Décomposition des noms des processus AGRIBALYSE et ecoinvent formatés pour SimaPro	32
9.6	Capture d'écran du calcul avec openLCA depuis MEANS-InOut	35

Liste des tableaux

8.1	Tableau comparatif des deux outils de calcul alternatifs à SimaPro	23
9.1	Extrait du mapping entre les catégories de flux élémentaires	30
9.2	Comparaison des résultats d'impacts pour des productions végétales simples	37

Extraits de code source et algorithmes

8.1	Exemple de fichier ecoSpold1	24
9.1	Connexion à une base de données Derby	27

9.2	Mapping d'un flux élémentaire	31
9.3	Import d'un objet ecoSpold1 dans openLCA avec mapping à la volée . . .	33

Première partie

**Présentation de l'UMR SAS et de
MEANS**

Chapitre 1

UMR SAS

1.1 Définition et composition

L'Unité Mixte de Recherche (UMR) Sol Agro et hydrosystème Spatialisation (SAS) repose sur l'Institut National de Recherche pour l'Agriculture, l'alimentation et l'Environnement (INRAE)¹ et sur le centre de Rennes de École nationale supérieure des sciences agronomiques, agroalimentaires, horticoles et du paysage (Agrocampus Ouest).

Composée d'environ une centaine d'enseignants-chercheurs, chercheurs, ingénieurs, techniciens, personnel administratif, stagiaires, doctorants et contractuels (CDD), elle se divise en deux sites : le premier (qui concentre une majorité du personnel) à Rennes dans les locaux d'Agrocampus Ouest, le second à Quimper.

Créée en 2000, l'UMR SAS est le résultat de la fusion de deux unités de l'INRA : « Sol et Bioclimatologie » et « Agronomie ». En 2004, l'unité « Physique des surfaces naturelles » de l'Agrocampus Ouest s'est jointe à l'UMR.

L'unité s'appuie aujourd'hui sur :

- trois départements de recherche INRAE à l'UMR SAS : « Environnement et Agronomie », « Physiologie Animale et Systèmes d'Élevage » et « Écosystèmes aquatiques, ressources en eau et risques » ;
- deux départements d'enseignement et de recherche à Agrocampus Ouest : « Milieu Physique, Paysage, Territoire » et « Sciences du Végétal pour l'Agriculture et l'Horticulture ».

1.2 Domaine de recherche

Les compétences de l'UMR SAS relèvent de l'agronomie, de la science du sol, de l'hydrologie et de l'évaluation environnementale. Ses travaux sont centrés sur les interactions entre l'agriculture et le milieu naturel par une approche intégrative et spatialisée des territoires ruraux (en France et à l'étranger). Les thématiques principales de recherche portent sur les cycles hydrologiques et biogéochimiques (azote, carbone, phosphore) dans les paysages cultivés, ainsi que l'évaluation multicritère des systèmes agricoles. Ces travaux visent à élaborer des systèmes de production agricole durables, à améliorer l'aménagement de l'espace rural, à préserver la qualité de l'eau, du sol, de l'air et des paysages.

¹Ex-INRA (Institut National de la Recherche Agronomique), depuis la fusion avec l'IRSTEA (Institut National de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture) depuis janvier 2020.

1.3 Organisation

1.3.1 Deux axes de recherche

Les recherches de l'UMR s'organisent en deux axes thématiques :

- Axe AGILE : accompagner la transition agroécologique des systèmes et territoires d'élevage ;
- Axe Processus : cycles et flux des éléments chimiques dans les agro-écosystèmes (p. ex. travail sur l'impact des effluents d'élevage).

En outre, un chantier dit « transversal » a été mis en place pour relier les travaux des deux axes de l'UMR, dans le but de développer des méthodes d'évaluation des services écosystémiques (p. ex. préservation de la biodiversité grâce aux haies en bordure de champs) rendus par les agrosystèmes à différentes échelles spatiales.

1.3.2 Quatre pôles techniques

L'appui technique aux travaux de recherche est réparti en quatre pôles :

- Pôle AnaTer : ANALyses, expérimentations, observations, TERrain ;
- Pôle Basmati : BASE de données, Modèles, Applications et Techniques Informatiques ;
- Pôle GesCom : GESTion COMmunication ;
- Service documentation : gestion d'un fonds documentaire spécialisé en agriculture, pédologie, hydrologie, agronomie et environnement.

L'UMR SAS héberge la plate-forme MEANS, au sein de laquelle le stage a été effectué.

Chapitre 2

MEANS

2.1 Définition

La plate-forme nationale MulticritEria AssessmeNt of Sustainability (MEANS) vise à mettre à la disposition de ses utilisateurs un ensemble d'outils informatiques pour l'analyse multicritère de la durabilité (sociale, économique et environnementale) de systèmes de production végétale, de production animale et de transformation des produits issus de l'agriculture.

Développée depuis 2012 par INRAE au sein de l'UMR SAS, et depuis 2018 avec le Centre de coopération internationale en recherche agronomique pour le développement (Cirad), MEANS a été ouverte à INRAE fin 2014, et à tout utilisateur (p. ex. cabinet de conseil en stratégie de développement durable) via des contrats de prestation de service, en janvier 2017.

MEANS fournit actuellement deux outils :

- l'application web MEANS-InOut qui permet la collecte des données décrivant un système de production animale ou végétale, le calcul flux généré par le système, et leur export dans un format interprétable par un logiciel d'Analyse du Cycle de Vie (ACV). Développée par l'équipe opérationnelle de MEANS ;
- un logiciel de calcul des indicateurs d'impacts environnementaux (SimaPro).

MEANS est partenaire du projet AGRIBALYSE de l'Agence de l'Environnement et de la Maîtrise de l'Énergie (ADEME), qui vise à produire les données de référence pour l'analyse environnementale des produits agricoles. La plate-forme implémente les outils permettant de générer et mettre à jour les bases de données qui sont accessibles aux partenaires d'AGRIBALYSE.

Au sein de l'UMR SAS, les membres de l'équipe de MEANS participent à l'Axe Agile et au pôle Basmati (cf. 1.3).

2.2 Gouvernance

MEANS est constituée de plusieurs instances présentées dans la figure 2.1.

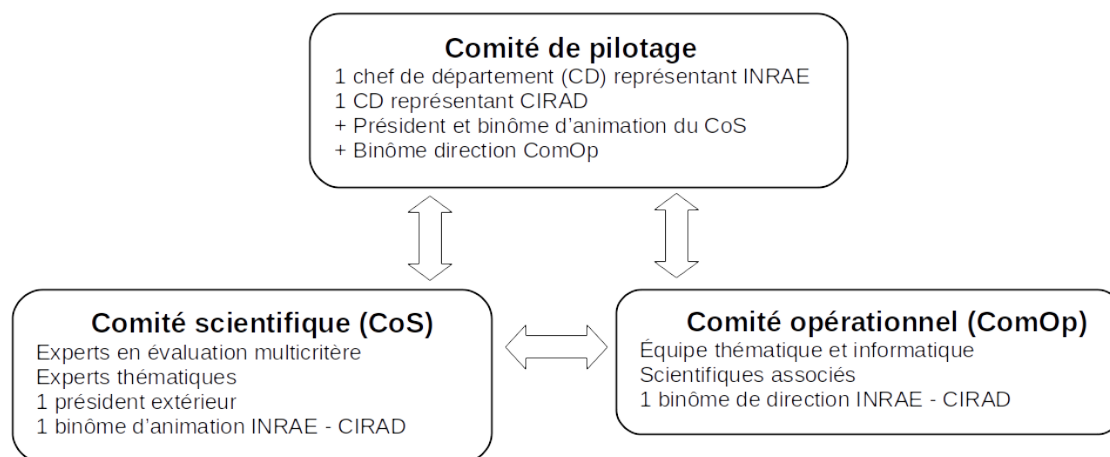


FIGURE 2.1 – Organigramme des différentes instances de la plate-forme MEANS

Source : Julie Auberger

Le stage s'est donc déroulé au sein de l'équipe informatique du comité opérationnel.

2.3 Fonctionnement de l'équipe

Le comité opérationnel est composé de :

- deux ingénieurs agronomes, dont le responsable de MEANS ;
- une chef de projet informatique ;
- deux ingénieurs en développement logiciel ;
- un assistant ingénieur, administrateur système et bases de données.

L'équipe fonctionne en mode AGILE, avec des sprints d'un mois environ, s'achevant par une réunion de démonstration. Une réunion quotidienne se déroulait également chaque matin avec les deux ingénieurs informatique et notre chef de projet, pour faire le point sur les avancées de la veille et discuter des développements de la journée.

2.4 MEANS-InOut

MEANS-InOut est une application web permettant de réaliser des évaluations environnementales de productions agricoles. L'objectif de ces évaluations est de contribuer à la conception de systèmes agricoles ayant moins d'impact sur l'environnement.

Pour ce faire, MEANS-InOut facilite la mise en œuvre de l'ACV pour les productions agricoles, en suivant la méthode AGRIBALYSE[7], co-construite par les experts des filières agricoles.

C'est le projet qui occupe la majeure partie du travail de l'équipe opérationnelle.

2.5 L'ACV et MEANS-InOut

L'ACV est une méthode d'évaluation normalisée (ISO 14044) permettant de réaliser un bilan environnemental multicritère quantitatif d'un produit ou d'un service. Cette méthode permet d'avoir une vision globale des impacts environnementaux d'un produit et d'en identifier les origines[7]. Une ACV se déroule en quatre étapes :

1. Définition des objectifs et du système. Par exemple : « on cherche à quantifier l'impact environnemental de la production de 20 t de maïs cultivé en agriculture biologique » ;
2. Inventaire des intrants et émissions, ou Inventaire de Cycle de Vie (ICV). Description de l'Itinéraire technique (ITK)[10, 11] et quantification de tous les flux (consommations de ressources et émissions de polluants) généré par le système ;
3. Évaluation de l'impact environnemental. Transformation de l'inventaire en indicateurs d'impacts (p. ex. changement climatique, eutrophisation, etc.) ;
4. Interprétation des résultats.

MEANS-InOut est utilisé pour les étapes 1 et 2 (cf. figure 2.2). Un logiciel d'ACV est requis pour l'étape 3. En l'occurrence, pour les utilisateurs de MEANS, il s'agit de SimaPro.

Les composants de l'inventaire (étape 2) sont issus de bases de données d'ACV, comme AGRIBALYSE[1] pour les composants spécifiques à l'agriculture. L'autre base

2.5. L'ACV ET MEANS-INOUT

utilisée par MEANS-InOut est ecoinvent[12]. Il s'agit de la principale base d'ACV, développée par l'entreprise du même nom. Les données qu'elle compile couvrent l'impact des métaux, matériaux de construction, produits chimiques, infrastructures, transports ou encore du traitement des déchets.

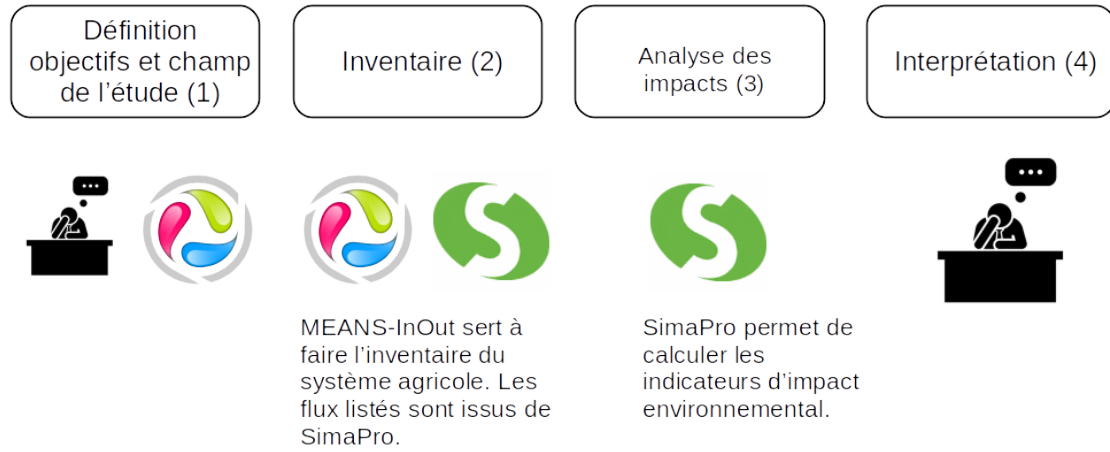


FIGURE 2.2 – Fonctionnement d'une analyse environnementale avec MEANS-InOut

Deuxième partie

Mission

Chapitre 3

Présentation du sujet

3.1 Motivations

3.1.1 Automatiser la production des résultats d'impacts

La plateforme MEANS permet de calculer l'impact environnemental de produits agricoles. Ce résultat s'obtient en quatre étapes (cf. figure 2.2), faisant intervenir deux outils :

1. Dans l'application web MEANS-InOut :
 - (a) **Saisie des itinéraires techniques.** L'utilisateur renseigne d'abord les caractéristiques de son ITK, par exemple l'ensemble des actions et des intrants mobilisés par un agriculteur pour une production d'une tonne de maïs. Il indique ainsi s'il y a un travail du sol (p. ex. labour), un épandage d'engrais, le carburant du tracteur, etc ;
 - (b) **Calcul des flux.** Les émissions (vers l'eau, le sol et l'air) et les consommations de ressources de l'ITK sont calculées suivant des modèles d'estimation ;
 - (c) **Export des données.** L'utilisateur obtient un fichier XML (cf. code 8.1) contenant l'ensemble des intrants (p. ex. de l'eau s'il a indiqué avoir irrigué sa parcelle) et émissions (p. ex. le dioxyde de carbone rejeté par le tracteur en fonctionnement) de son ITK. On parle d'un ICV. Les données de l'ITK sont ainsi « mappées » vers des intrants et émissions issus de bases de données d'ACV ;
2. Dans le logiciel de calcul d'impacts (SimaPro) :
 - (a) **Calcul des indicateurs d'impacts.** L'utilisateur importe le fichier XML obtenu en sortie de MEANS-InOut. Il peut alors lancer le calcul des indicateurs d'impacts de son ICV, qu'il obtiendra dans un écran du logiciel et pourra copier s'il le souhaite vers un tableur pour les exploiter.

Le problème réside dans le caractère manuel du lien entre les deux outils. Cette limite est due au fait que SimaPro ne propose pas d'API (Application Programming Interface) facile d'utilisation, ou tout du moins pas faite pour une application Java, comme MEANS-InOut¹. À défaut, et à condition que le code source soit ouvert, un logiciel tiers comme MEANS-InOut pourrait développer lui-même cette interface. Or, SimaPro étant un logiciel propriétaire, cette solution est inenvisageable.

¹<https://support.simapro.com/articles/Manual/Programming-the-COM-Interface>

La figure 3.1 résume l'objectif : générer directement les résultats d'impacts depuis MEANS-InOut, en utilisant un autre outil de calcul que Simapro.

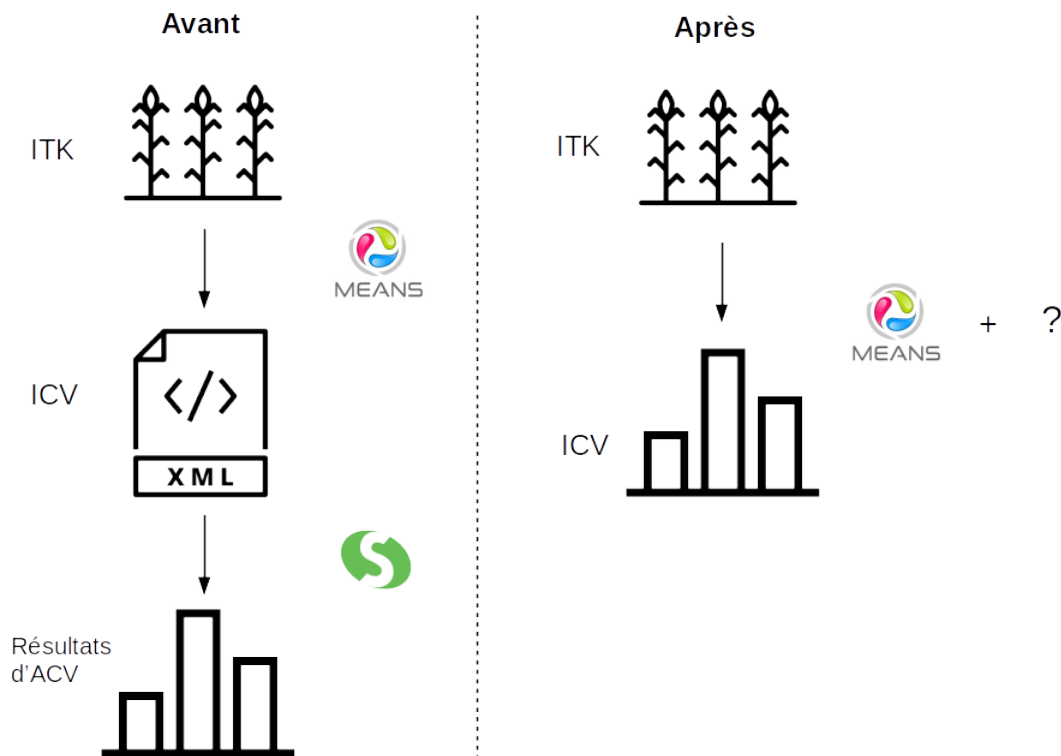


FIGURE 3.1 – Objectif principal du stage

3.1.2 Pouvoir lancer des calculs en série

Au-delà du gain de temps que permettrait la suppression de l'étape manuelle d'import, le fait de pouvoir programmatiquement exécuter des calculs d'impacts ouvrirait la possibilité de lancer des calculs en série. Soit sur plusieurs ITK, soit sur le même mais en faisant varier un paramètre, pour réaliser des études de sensibilité (p. ex. s'intéresser à la variation des impacts de la production d'une tonne de maïs en fonction du nombre d'heures de travail mécanique).

Le calcul en série pourrait intéresser des projets dont l'équipe MEANS est partenaire. C'est par exemple le cas d'un projet porté par le Système d'information (SI) INRAE AgroSyst', qui compile déjà plusieurs milliers d'ITK de productions végétales. MEANS-InOut permettrait d'ajouter des indicateurs au calcul de leur impact.

3.1.3 Étendre l'utilisation de MEANS-InOut

SimaPro étant un logiciel propriétaire, il a un coût. Dès lors, l'automatisation des calculs d'impacts est l'occasion de se tourner vers une alternative open-source. Cette perspective intéresse des chercheurs du Cirad, partenaire de MEANS. Travaillant notamment dans les pays en développement, ils n'ont pas toujours les fonds nécessaires pour acquérir un logiciel comme SimaPro.

Le fait de s'affranchir de SimaPro rendrait également l'utilisation de MEANS plus accessible pour des utilisateurs non spécialistes de l'ACV.

3.2 Sujet initial et évolutions

3.2.1 Choisir une solution de calcul alternative

Le premier objectif du stage est de retenir un outil pour remplacer SimaPro, qui permette d'exécuter des calculs d'impacts programmatiquement, depuis MEANS-InOut. Deux outils open-source ont déjà été identifiés par l'équipe MEANS. Le premier, Brightway2[8], est une librairie Python. Le second, openLCA[4] est un logiciel de bureau, reposant sur un ensemble de modules[3] Maven (développés en Java).

3.2.2 Connecter la solution choisie à MEANS-InOut

Il s'agit ici d'implémenter la communication entre les deux outils : le premier fournit un ICV, le second calcule des résultats d'impacts. Le point de départ est le module d'export existant, qui produit un fichier XML. Cette tâche, objectif principal du stage, a été réalisée.

L'utilisateur peut alors modifier son ITK et obtenir les résultats d'impacts mis à jour, sans passer par un logiciel d'ACV. MEANS-InOut se rapproche ainsi d'un de ses objectifs initiaux, à savoir rendre accessible l'ACV à des non-spécialistes[2]. Un utilisateur plus avancé pourrait souhaiter re-travailler l'ICV dans un logiciel dédié, comme c'était le cas jusqu'à présent avec SimaPro. Cette fonctionnalité n'a pas été développée mais peut l'être rapidement car la solution choisie offre une fonction d'export vers plusieurs formats, qui peuvent ensuite être importés dans la version bureau de l'application.

3.2.3 Développer des interfaces de visualisation des impacts

L'analyse des impacts d'un ITK permet de répondre à plusieurs questions : quelle est la contribution des différentes étapes à l'impact total? (p. ex. est-ce l'épandage des engrais ou le travail du sol qui impacte le plus la qualité du sol?), y'a-t-il des transferts d'impact? (p. ex. un épandage d'engrais qui réduirait l'impact sur la qualité de l'air, mais aggraverait celui sur la qualité de l'eau), etc.

Pour répondre à ces questions, les outils d'ACV présentant une interface (p. ex. SimaPro et openLCA) proposent différents graphes de visualisation des impacts. En obtenant ces derniers directement depuis MEANS-InOut, il devient possible d'y intégrer de telles visualisations.

Si ces graphes offrent un premier aperçu des différentes tendances des impacts, ils sont néanmoins souvent incomplets pour les praticiens de l'ACV. De telle sorte que ces derniers préfèrent souvent copier les résultats dans un tableur pour les traiter eux-mêmes et éventuellement produire les graphes qui les intéressent. Ce point a été soulevé dans la première moitié du stage, par l'agronome de l'équipe MEANS.

Dès lors, cette tâche est apparue non pertinente, ou du moins non prioritaire du point de vue des besoins des utilisateurs de MEANS.

3.2.4 Réaliser des calculs en série

Si le calcul d'impacts devient une fonctionnalité de l'outil MEANS-InOut, alors il devient possible de calculer en une fois les impacts d'un ensemble d'ITK. Ces calculs n'étant pas instantanés, ils devraient être exécutés par lot (batch). De tels calculs en série permettent alors de comparer les impacts de plusieurs ITK, comme évoqué en 3.1.2.

Cette tâche n'a pas été instruite, car d'autres problématiques jugées plus prioritaires sont apparues, notamment la vérification de l'exactitude des résultats obtenus avec l'outil de calcul choisi et le travail sur l'exploitabilité de ces résultats.

3.2.5 Exporter les résultats d'impacts vers une base de données sémantique

Une des initiatives à l'origine de l'idée d'automatiser le calcul d'impacts environnementaux dans MEANS-InOut est le projet DataSusFood (*Structuring and Opening Data to improve Sustainability of Food Systems*) [9], financé par l'Agence nationale de la recherche (ANR) pour deux ans (2020-2021). En s'appuyant sur plusieurs travaux existants, dont MEANS-InOut, ce projet vise à produire des données respectant les principes FAIR (*Findable, Accessible, Interoperable, Reusable*) sur l'analyse multicritère de systèmes agro-alimentaires.

Dans ce cadre, une des étapes consiste à rendre interopérables MEANS-InOut et la base de données BaGaTel [5], qui renseigne sur les produits alimentaires et leurs processus de transformation. En particulier, le projet prévoit le développement par MEANS-InOut d'un module permettant d'exporter les résultats d'impacts vers BaGaTel. Le stage a permis d'implémenter le calcul d'impacts, il reste donc à développer l'export de ces données vers BaGaTel.

Chapitre 4

Contributions

Le stage se partageait a priori en deux grandes phases : d'abord implémenter le calcul d'impacts via un autre outil que SimaPro, puis exploiter cette nouvelle fonctionnalité. Plusieurs inconnues rendaient peu évidente une planification a priori, mais on pouvait estimer que l'exploitation du service de calcul pourrait commencer pendant le dernier tiers du stage.

Dans les faits, l'interopérabilité entre MEANS-InOut et l'outil de calcul retenu s'est avérée moins évidente qu'espéré. Un travail supplémentaire sur l'exploitabilité des résultats d'impacts a également été nécessaire. De telle sorte que la deuxième partie du stage a finalement consisté à rendre le service de mapping (cf. section suivante) plus extensible et le service de calcul plus robuste.

4.1 Outil de mapping entre MEANS-InOut et openLCA

La première contribution du stage est un algorithme permettant de faire correspondre les bases de données de MEANS-InOut et openLCA (le choix de cet outil sera justifié par la suite). Pour rappel, MEANS-InOut produit un ICV dont openLCA calcule les

impacts. Pour ce faire, les intrants et émissions listés dans l'ICV doivent pouvoir être retrouvés dans la base de données de openLCA.

Or, les identifiants d'origine de ces flux (intrants et émissions) ne sont pas exploités par les logiciels d'ACV et le format de l'ICV ne les supporte pas. En conséquence, les flux sont identifiés par leurs libellés, lesquels dépendent des logiciels d'ACV, qui ne conservent pas exactement les valeurs de la base originale. Les données de MEANS-InOut étaient ainsi conformes à celles de SimaPro. Il fallait donc les « dé-formater » pour les rendre cette fois compatibles avec openLCA.

Un algorithme dit de *mapping* (au sens où il fait correspondre des données entre elles) a donc été développé pour que les impacts des ICV produits par MEANS-InOut puissent être calculés par openLCA. Cet outil a été conçu aussi générique que possible, pour pouvoir supporter les mises à jour (fréquentes) des bases de données d'ACV.

4.2 Outil de calcul d'impacts via openLCA

L'autre contribution réside dans la possibilité désormais offerte à un utilisateur d'obtenir les impacts de son ITK directement depuis MEANS-InOut. Cette fonctionnalité a été testée sur les productions végétales, mais pas sur les productions animales ni piscicoles.

L'implémentation du calcul à proprement parler n'a pas été la phase la plus chronophage, consistant simplement à appeler les fonctions déjà définies par openLCA. C'est davantage la comparaison des résultats avec ceux obtenus via SimaPro qui a pris du temps.

Chapitre 5

Planning

5.1 Déroulement effectif du stage

En début de stage, du fait de l'inconnue sur l'outil de calcul et sur les éventuelles difficultés d'intégration avec MEANS-InOut, il n'avait pas été établi de planning détaillé des étapes à suivre. Néanmoins, il avait tout de même été fixé que le premier mois serait consacré à la prise en main de MEANS-InOut et au choix de l'outil de calcul et qu'un point d'étape serait fait à mi-stage sur l'avancement de la connexion entre les deux outils. Comme expliqué en section 4, la suite du stage n'a pas été orientée vers l'exploitation du module de calcul comme prévu initialement, mais a plutôt été consacrée à son perfectionnement. Le planning suivi (sous une forme résumée) a donc été le suivant :

- Lancement du stage (1 mois) :
 - prise en main de MEANS-InOut ;
 - choix de l'outil de calcul et du format d'échange de données.

- 1^{re} phase de développement (trois mois) :
 - intégration des bibliothèques d’openLCA dans le projet MEANS-InOut, développement de la connexion à une base de données openLCA (1,5 sem.);
 - mapping entre les données de MEANS-InOut et openLCA (3 sem.);
 - implémentation du calcul d’impacts pour des productions végétales simples (1 sem.);
 - comparaison des résultats avec SimaPro;
 - analyse et correction du mapping après constat de résultats d’impacts différents (3 sem.);
 - fusion avec branche principale et déploiement sur le serveur de développement;
 - refactoring et nettoyage du service de calcul;
- 2^e phase de développement (2 mois) :
 - amélioration de l’UX pour lancer les calculs;
 - amélioration des temps de calcul;
 - documentation du service de mapping et présentation du travail réalisé depuis le début du stage;
 - amélioration du service de mapping, correction de bugs et nettoyage du code;
 - ajout du calcul d’impacts de successions de cultures et comparaison des résultats avec SimaPro;
 - rédaction du rapport.

5.2 Étapes principales

Le cœur du stage résidait a priori dans l’utilisation des méthodes de calcul d’openLCA. Finalement, comme déjà dit en section 4, une étape non négligeable de mapping entre les bases de données de MEANS-InOut et openLCA a été nécessaire.

La partie la plus intéressante, mais aussi la plus fastidieuse et chronophage, a sans doute été celle du mapping. Elle a commencé par une prise de conscience des problèmes d’interopérabilité des données en Analyse du cycle de vie (qui n’est pas propre à ce domaine). Il a alors fallu analyser ces données, identifier les différences de format, pour développer des stratégies de mapping. Lesquelles devaient non seulement fonctionner pour les bases existantes, mais aussi futures.

Une fois ces problèmes d’interopérabilité écartés, l’exploitation des fonctions de calcul d’openLCA a bien-sûr été l’autre étape importante du stage. Quoique moins difficile à mettre en œuvre, elle a nécessité plusieurs échanges avec différents acteurs, qui l’ont rendue intéressante. D’une part avec les développeurs d’openLCA, grâce au caractère open-source de leur outil. D’autre part avec les chercheurs de MEANS, sur le niveau et la nature des informations apportées par les résultats d’impacts, pour correspondre à

leurs besoins.

Chapitre 6

Outils et technologies

6.1 Développement

MEANS-InOut est une application web Java développée selon le framework Spring MVC¹. Le traitement métier des données est réalisé dans des « services » (classes Java) qui ont la responsabilité d'une action précise nécessaire aux contrôleurs (p. ex. calculer l'impact d'un ICV donné). Pour le stage, l'essentiel des développements a donc été fait dans cette couche métier.

Les modules d'openLCA utilisés dans le cadre du stage ont été intégrés à MEANS-InOut en tant que nouvelles dépendances, via le gestionnaire de dépendances Maven. Le code d'openLCA consistait lui aussi en classes Java.

La base de données de MEANS-InOut est une base relationnelle PostgreSQL. Un travail d'analyse de certaines tables a été nécessaire au cours du stage, notamment pour le mapping. Quant à openLCA, deux technologies de bases de données étaient possibles : Apache Derby et PostgreSQL. Il a été choisi d'exploiter la première solution car c'est le format utilisé par la version bureau du logiciel, ce qui permettait de ne pas avoir à créer soi-même la base de données. Cette dernière est ainsi stockée dans un fichier local.

Le versionnage de MEANS-InOut est réalisé avec Subversion (SVN). La première phase de développement (cf. 5.1) a été réalisée dans une branche distincte de la branche principale. Cependant, contrairement à Git, SVN n'encourage pas à travailler avec plusieurs branches. De fait, l'équipe de MEANS-InOut fonctionnait uniquement avec la branche principale (le *trunk*) La deuxième phase de développement a donc suivi ce principe.

Le stage a également permis de se familiariser avec le serveur d'application Payara, sur lequel est déployée MEANS-InOut, ainsi qu'avec l'outil d'intégration continue Jenkins.

6.2 Organisation du travail

L'équipe de MEANS-InOut travaille selon la méthode Agile. Les tâches sont gérées avec le logiciel Jira, une réunion d'équipe se tient chaque matin et les sprints durent environ un mois. Pour le stage, un kanban (tableau de gestion de l'avancement des tâches) dédié a été créé dans Jira.

¹<https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html>

6.3 Tests

Les développements réalisés pour MEANS-InOut sont testés avec le framework de test unitaire JUnit. Pour le stage, il est apparu peu évident d'écrire des tests unitaires. En effet, le module de calcul mobilisait deux bases de données : celle de MEANS-InOut et celle d'openLCA. Dès lors, il aurait fallu créer autant de mocks (objets simulés) pour les tests.

Le choix a donc été fait de tester « manuellement » le module de calcul, en comparant les résultats avec ceux obtenus via l'ancienne solution (SimaPro). Ces tests ont été réalisés tout au long du développement, et validés avec l'agronome de MEANS. Une tolérance de 5% de différence a été fixée, sachant qu'en ACV, il est courant que deux logiciels de calcul obtiennent des résultats non strictement égaux, ne serait-ce que pour des différences d'arrondis qui additionnés les uns aux autres, finissent par créer un delta.

Chapitre 7

Prise de recul

L'objectif principal du stage a été atteint : développer un module de calcul d'impacts au sein de MEANS-InOut. Les résultats peuvent désormais s'obtenir directement depuis l'application elle-même. Il n'est donc désormais plus nécessaire (pour les productions végétales seulement pour l'instant) de disposer d'un logiciel tiers (SimaPro) pour bénéficier pleinement des services de MEANS. Cette avancée ouvre plusieurs perspectives, en particulier le calcul d'impacts en série, des études de sensibilité plus rapides à mettre en œuvre, l'extension de l'utilisation de l'application et la collaboration avec d'autres projets (cf. 3.2.5). Le lien établi avec openLCA ouvre également la possibilité d'utiliser davantage des fonctionnalités de cet outil, en particulier la spatialisation des impacts, et l'intégration de la dimension sociale voire économique (les deux autres dimensions de la durabilité avec l'environnement) dans les indicateurs d'impacts.

Plusieurs améliorations sont envisageables. En premier lieu, améliorer la configuration de calcul pour qu'elle corresponde au mieux aux besoins des utilisateurs de MEANS-InOut. Par ailleurs, le module de calcul est appelé à être étendu aux productions animales. Le nécessaire a été fait pour que le module puisse supporter les futures versions des bases de données d'ACV. Enfin, et même si ce travail apparaît moins prioritaire, il serait sans-doute bénéfique de ne plus passer par le format ecoSpold1, mais de directement utiliser le format développé par openLCA, JSON-LD (cf. 8.2.3).

Troisième partie

Réalisations

Chapitre 8

Lancement du stage

8.1 Choix de l’outil de calcul

8.1.1 Travail antérieur au stage

Le sujet du stage a été créé pour concrétiser un travail déjà amorcé depuis quelques années par l’équipe de MEANS. La possibilité d’une refonte de l’export au format ecoSpold1 pour le rendre compatible avec openLCA avait ainsi été étudiée en 2015. Début 2019, c’est l’alternative d’un nouvel export, cette fois-ci vers le format ecoSpold2, mais toujours dans l’optique d’une intégration avec openLCA, qui avait été explorée.

Ces tentatives successives illustrent une difficulté importante en ACV : les changements de versions des bases de données et des formats de description d’ICV qui les accompagnent, qui limitent l’interopérabilité entre les différents outils. L’application MEANS-InOut a été créée en 2012, avec la version d’alors d’ecoinvent (v2) et le format d’ICV correspondant, ecoSpold1. Dès 2013, ecoinvent livre une nouvelle version de sa base de données¹. Un des changements principaux est l’ajout d’un identifiant dans le XSD (XML Schema Definition) d’un processus. En conséquence, une nouvelle version du format ecoSpold accompagnait ecoinvent 3.0.

Cette difficulté s’ajoute à la variété des autres formats d’ICV, notamment dû au fait qu’un logiciel comme SimaPro n’utilise pas le format ecoSpold standard, mais une version qu’il a adaptée à ses besoins. D’où la nécessité d’un « mapping », que ce soit pour convertir un ICV d’un format à un autre, ou migrer une base de données vers la version supérieure. Une entreprise comme openLCA, entièrement dédiée à son logiciel d’ACV, disposait du temps et des ressources pour développer, dès 2006, un convertisseur de formats² (qui ne gérait cependant pas le problème d’incohérence des libellés, cf. 4.1). C’est moins évident pour une unité publique et de moindre effectif comme celle de MEANS-InOut.

Le constat peut être fait que le travail exploratoire réalisé de façon ponctuelle avant le stage s’était concentré sur openLCA, qui existe depuis 2006. En 2012, un autre outil, Brightway, a été créé dans le cadre d’un post-doctorat. D’autres outils d’ACV existent, mais ne sont pas open-source (cf. 3.1.3). Ce sont ces deux outils (openLCA et Brightway) qui étaient donc proposés à l’étude dans le sujet de stage.

¹<https://www.ecoinvent.org/database/older-versions/ecoinvent-30/ecoinvent-30.html>

²<http://www.openlca.org/format-convertter>

8.1.2 Tests d'openLCA et Brightway

Critères de choix

D'une part, la solution retenue doit pouvoir s'intégrer facilement avec MEANS-InOut, application Java et qui gère ses dépendances avec Maven. D'autre part, elle doit proposer une version bureau avancée et stable, afin d'envisager une alternative graphique à SimaPro pour les utilisateurs de MEANS en particulier, et pour les membres de l'UMR SAS réalisant des ACV en général. On veut en effet éviter une situation où l'UMR aurait migré vers un logiciel d'ACV différent de celui utilisé par MEANS-InOut. Ceci éviterait les différences de résultats entre logiciels (cf. 6.3) et faciliterait la possibilité d'exporter un ICV depuis MEANS-InOut pour le retravailler dans la version bureau de l'outil (cf. 3.2.2).

Brightway

Brightway est une librairie Python. A priori, l'utilisation de cette dernière ne peut donc se faire au sein du code de MEANS-InOut, mais plutôt dans une application serveur dédiée, dont MEANS-InOut consommerait l'API. Le test de la librairie s'est limité à un travail réalisé peu avant le stage par un membre de l'équipe MEANS, qui tentait (avec succès) d'importer la base de données AGRIBALYSE.

En complément, les deux interfaces graphiques développées à ce jour ont été testées. La première, Activity Browser³, est assez avancée. Néanmoins, elle ne peut encore s'installer qu'en ligne de commande, confirmant cette limite assumée sur le site même de Brightway : « So it's not for everyone, and you will need to invest some time learning about Python to use Brightway2. » [8] La seconde, lcopt⁴, est moins mature et là aussi pour l'instant uniquement destinée à des utilisateurs à l'aise en informatique. En outre, lcopt propose une approche beaucoup plus graphique que SimaPro, ce qui pourrait freiner son adoption. Enfin, l'application revendique clairement avoir été conçue pour des ACV avec paramètres, ce dont MEANS-InOut n'a pas eu besoin jusqu'alors.

Ce choix de mettre l'accent sur une dimension avancée de l'ACV confirme la philosophie de Brightway : « It doesn't try to replace software like SimaPro or OpenLCA, but rather to offer new possibilities to those who want to break the limits of conventional LCA. (...) Basically, this is the framework for you if your project lies at the intersection of LCA and your imagination. » [8]

openLCA

openLCA est un logiciel reposant sur un ensemble de modules Maven, développés en Java[3]. Bien que déjà éprouvée par la communauté ACV, son interface a été testée. Il en est ressorti que son adoption ne posait pas de problème pour qui est déjà habitué à un logiciel comme SimaPro. Cette impression a été confirmée lors d'un entretien avec Lynda Assani, ingénieure de recherche en évaluation environnementale, qui utilise openLCA pour des cours et des travaux pratiques avec ses étudiants.

Restait ensuite à vérifier l'intégration avec MEANS-InOut. Un premier test a été réalisé avec la librairie Python olca-ipc⁵. Cette dernière permet de communiquer avec un serveur IPC (Inter-Process Communication) donnant accès à une base de données openLCA créée avec la version bureau du logiciel. Il faut donc préalablement démarrer

³<https://github.com/LCA-ActivityBrowser/activity-browser>

⁴<https://lcopt.readthedocs.io>

⁵<https://olca-ipc.readthedocs.io>

le serveur via l'interface graphique, frein évident à un calcul automatisé. Néanmoins, l'objectif était davantage de se familiariser avec le modèle de données d'openLCA. Ces expérimentations ont fait apparaître les limites de l'API d'olca-ipc. À titre d'exemple, pour récupérer un processus d'après son nom et sa localisation (p. ex. la production de fioul en France), il fallait récupérer l'ensemble des processus et filtrer soi-même, ce qui posait des problèmes de performance.

L'autre solution était d'utiliser directement le module Maven permettant de se connecter à une base de données. Cette alternative, la plus simple, et qui s'est avérée fonctionnelle, permettait également d'accéder à l'ensemble du code métier d'openLCA, et donc de s'affranchir des limites d'olca-ipc. En outre, MEANS-InOut utilisait déjà un module d'openLCA pour créer ses fichiers ecoSpold1. La démarche était donc éprouvée.

Un autre atout réside dans le support assuré par openLCA, via un forum dédié⁶.

8.1.3 Solution retenue

Le choix s'est finalement porté sur openLCA. Le tableau 8.1 récapitule les différences avec Brightway.

	openLCA	Brightway
Maturité	Développé depuis 2007	Développé depuis 2012
Intégration avec MEANS-InOut	Modules indépendants écrits en Java, même langage de programmation que MEANS-InOut.	Librairie écrite en Python, plus difficilement intégrable avec MEANS-InOut.
Support technique	Forum de questions-réponses, équipe technique dédiée.	Un développeur principal.
Autres	Interface utilisateur mature, support des impacts spatialisés, mode collaboratif : intéressant à terme pour MEANS voire UMR SAS.	Librairie puissante, mais encore uniquement dédiée aux développeurs.

TABLE 8.1 – Tableau comparatif des deux outils de calcul alternatifs à SimaPro

Comme dit précédemment, MEANS-InOut génère ses ICV au format ecospold1 en utilisant un module d'openLCA. La solution évidente était d'exploiter ces objets pour les créer dans une base de données openLCA, et exécuter les calculs. Néanmoins, d'autres formats de données étaient possibles, en particulier ecoSpold2. Un autre travail de sélection a donc dû être effectué pour choisir le format le plus pertinent.

8.2 Choix du format d'échange de données

8.2.1 Fonction du format d'échange

Un ICV doit pouvoir être décrit suivant un format standard pour être interprétable par le logiciel d'ACV qui doit calculer ses impacts. Par exemple, si un utilisateur a décrit l'itinéraire technique d'un ensilage de maïs, le fichier ecoSpold1 généré par MEANS-InOut se présente comme suit (version simplifiée) :

⁶<https://ask.openlca.org/>

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ecoSpold xmlns="http://www.EcoInvent.org/EcoSpold01">
3   <dataset number="1">
4     <metaInformation>
5       <processInformation>
6         <!-- le processus modélisé : la production de 20 T de maïs -->
7         <referenceFunction name="Silage maïze production"
8           ↪amount="20000.0" unit="kg" category="Agricultural"
9           ↪subCategory="Plant production\Forages"/>
10        <geography location="FR"/>
11      </processInformation>
12    </metaInformation>
13    <flowData>
14      <!-- le produit ("output") principal du processus : 20 T de maïs
15      ↪-->
16      <exchange number="1" name="Silage maïze" location="FR" unit="kg"
17        ↪meanValue="20000.0">
18        <outputGroup>0</outputGroup>
19      </exchange>
20      <!-- un intrant ("input") du processus : un engrais -->
21      <exchange number="2" name="Engrais" location="FR" unit="kg"
22        ↪meanValue="20">
23        <inputGroup>5</inputGroup>
24      </exchange>
25      <!-- une émission ("output") du processus : du phosphate -->
26      <exchange number="3" category="Water" subCategory="groundwater"
27        ↪name="Phosphate" location="FR" unit="kg" meanValue="20">
28        <outputGroup>4</outputGroup>
29      </exchange>
30      <!-- autres intrants et émissions... -->
31    </flowData>
32  </dataset>
33 </ecoSpold>

```

Code 8.1 – Exemple de fichier ecoSpold1

Les intrants et émissions (nous parlerons de « flux ») sont issus de bases de données d'ACV, essentiellement de ecoinvent (une des plus complètes, couvre tous les secteurs d'activité) et AGRIBALYSE (centrée sur les produits agricoles et alimentaires) pour MEANS. Le fichier ecoSpold référence donc ces flux, pour que le logiciel d'ACV récupère leurs impacts dans les bases, et calcule l'impact total du processus modélisé (cf. figure 2.2).

8.2.2 ecoSpold2

Le format ecoSpold2 a été introduit par ecoinvent avec la version 3 de sa base de données (cf. 8.1.1). Un des principaux changements est l'ajout dans la balise `exchange` de l'identifiant du flux dans sa base de données. Une modification a priori importante, dans la mesure où elle rend beaucoup plus évidente l'interopérabilité entre logiciels utilisant la même base de données.

Néanmoins, les bases de données d'ACV ne sont pas reprises telles quelles par les logiciels. Ainsi, openLCA (tout comme SimaPro) ne conserve pas l'identifiant des flux, et re-génère ses propres identifiants. MEANS-InOut n'a pas non plus stocké cette information, du fait que ecoSpold1 ne l'utilisait pas. Dès lors, le format ecoSpold2 ne s'avère plus d'une grande utilité.

Pourtant, openLCA avait commencé à supporter l'import de fichiers ecoSpold2, identifiant les flux comme pour ecoSpold1, c'est-à-dire en combinant les informations dispo-

nibles (le nom, la catégorie, l'unité, etc.). Cette fonctionnalité a cependant été supprimée. openLCA a expliqué sur son forum de support que le format manquait de documentation, et que la base de données ecoinvent, sensée respecter ecoSpold2, n'était en réalité pas exactement conforme au schéma décrit dans les spécifications.

8.2.3 Choix d'ecoSpold1

D'autres formats existent, notamment un développé par openLCA⁷, implémentant la norme JSON-LD (JavaScript Object Notation for Linked Data)⁸, spécifiée par le W3C. Même s'il ne l'exploite pas pour l'instant, ce format ouvre les données d'ACV au web sémantique.

Le choix s'est cependant tourné vers ecoSpold1, principalement parce que MEANS-InOut générerait déjà ce format. Choisir JSON-LD aurait impliqué d'implémenter la conversion avant de se pencher sur les calculs d'impacts. Pour autant, ecoSpold1 impliquera lui aussi quelques traitements préalables, en particulier parce que les valeurs des attributs (nom, catégorie, etc.) correspondent aux valeurs dans SimaPro, qui a modifié celles présentes dans les bases (ecoinvent, AGRIBALYSE) d'origine.

Chapitre 9

Développement

Les deux phases de développement décrites dans le planning (cf. section 5.1) ont ici été rassemblées en une seule partie, pour privilégier la présentation par fonctionnalité plutôt que par chronologie. Les tâches minimales, comme l'ajout d'éléments d'interface nécessaires pour le lancement des calculs, la correction de bugs et l'amélioration du code développé ne seront pas détaillées, pour se concentrer sur l'essentiel du travail réalisé : les modules d'import, de mapping, de calcul et la vérification des résultats.

9.1 Préparation de l'import des objets ecoSpold1

9.1.1 Intégration de la librairie d'import

La première étape du développement consiste à importer les objets ecoSpold1 générés par MEANS-InOut dans une base de données openLCA. Autrement dit, on doit donc « convertir » un ICV au format ecoSpold1 vers le modèle de données d'openLCA. Ce travail aurait pu être effectué manuellement, mais openLCA a déjà développé un module Maven dédié à ce type de conversion : `olca-io[3]` (pour « input output »). Il suffit donc d'ajouter ce module dans les dépendances de MEANS-InOut.

⁷<http://greendelta.github.io/olca-schema>

⁸<https://json-ld.org/>

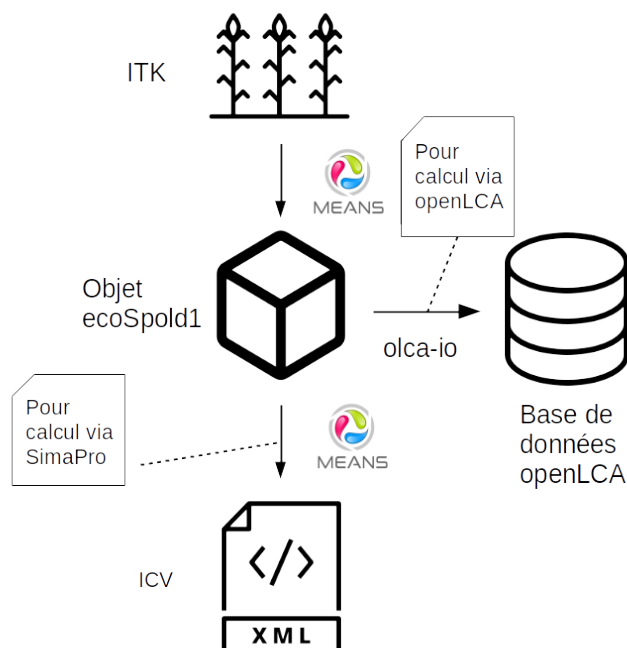


FIGURE 9.1 – Ajout de l’import dans openLCA au traitement d’un ITK par MEANS-InOut

openLCA utilise ce module pour gérer les nombreux formats d’ICV (cf. section 8.1.1). Par exemple, le module possède une méthode publique qui reçoit en entrée un fichier ecoSpold1 (cf. code 8.1), et écrit en base de données les différents objets contenus dans le fichier (intrants, émissions, etc.).

Cette méthode procède en deux temps : parsing du fichier pour créer un objet ecoSpold1, puis création en base des objets métier contenus dans l’ecoSpold1. Dans le cas de MEANS-InOut, la première étape n’est pas nécessaire. Il a donc fallu rendre publique la méthode de olca-io qui reçoit directement en entrée un objet ecoSpold1. La figure 9.1 illustre l’intégration de olca-io dans le flux d’opérations existant de MEANS-InOut.

L’import étant rendu possible, reste à choisir le système de gestion de base de données (SGBD) pour openLCA, entre PostgreSQL et Apache Derby.

9.1.2 Choix du SGBD pour openLCA

Le SGBD utilisé par la version bureau d’openLCA est Apache Derby. En effet, même s’il supporte le fonctionnement client-serveur classique, Derby a la particularité de pouvoir être embarqué dans une application Java, grâce à sa petite taille (2MB) et au fait que son code est écrit en Java. On peut donc directement l’utiliser dans une application Java, via son module Maven public¹.

openLCA propose également un script pour créer sa base de données sur un serveur PostgreSQL. Néanmoins, Derby a été retenu pour MEANS-InOut. D’une part pour éviter de devoir créer un serveur de base de données, et d’autre part car c’est le SGBD utilisé

¹<https://mvnrepository.com/artifact/org.apache.derby>

par la version bureau, ce qui allait faciliter les tests. En effet, on peut lancer l'import d'un ICV depuis MEANS-InOut, puis directement vérifier via l'interface graphique que tous les éléments ont bien été créés.

La base de données créée par l'application bureau se présente sous la forme d'un dossier directement stocké sur le poste de l'utilisateur. Le code 9.1 montre comment l'utiliser.

```

1 import java.io.File;
2
3 import org.openlca.core.database.IDatabase;
4 import org.openlca.core.database.derby.DerbyDatabase;
5 import org.openlca.core.database.ExchangeDao;
6 import org.openlca.core.model.Exchange;
7
8 public class DatabaseExample {
9     public static void main(String[] args) {
10         String dbPath = "C:/Users/test/openLCA-data/databases/ecoinvent_34";
11         IDatabase db = new DerbyDatabase(new File(dbPath));
12
13         Exchange exchange = new ExchangeDao(db).getForRefId(
14             "7d1cbce0-b5b3-47ba-95b5-014ab3c7f569");
15         System.out.println(exchange.flow.name);
16     }
17 }

```

Code 9.1 – Connexion à une base de données Derby

Cependant, pour pouvoir faire les vérifications dans l'interface, la connexion créée par MEANS-InOut devra avoir été fermée. En effet, s'il supporte la connexion concurrente de plusieurs utilisateurs d'une même application, le driver embarqué Derby autorise une seule application à être connectée à une même base, à un instant donné. C'est ce qu'illustre la figure 9.2.

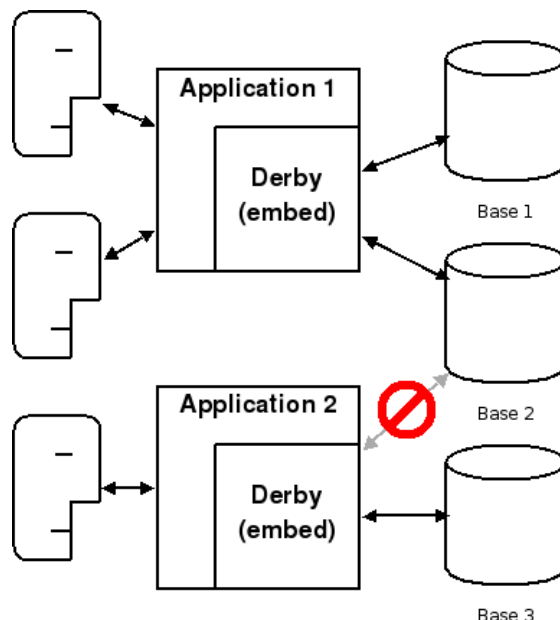


FIGURE 9.2 – Fonctionnalités d'un driver embarqué Derby

Source : http://www.chicoree.fr/w/Premiers_pas_avec_Apache_Derby

Pour MEANS-InOut, la base de données est donc d'abord créée depuis l'application bureau de openLCA, en y important les différentes bibliothèques d'ACV nécessaires (p. ex. ecoinvent, AGRIBALYSE). Une documentation a été rédigée pour détailler cette procédure. Elle est ensuite exploitée par MEANS-InOut suivant l'exemple du code 9.1.

L'import d'un objet ecoSpold1 est maintenant possible. Néanmoins, pour que ce dernier soit exhaustif, il s'est avéré nécessaire de « retravailler » l'ecoSpold1.

9.2 Mapping entre MEANS-InOut et openLCA

9.2.1 Préambule

Nécessité du mapping

Après quelques tentatives d'import, il a été constaté que très peu de flux référencés dans les ecoSpold1 produits par MEANS-InOut étaient « retrouvés » dans openLCA. En effet, comme expliqué en section 8.1.1, SimaPro effectue des modifications sur les bases d'ACV d'origine, ce qui avait obligé MEANS-InOut à s'y conformer. En conséquence, un travail de « dé-formatage » s'est avéré nécessaire.

La figure 9.3 illustre ce travail de « mapping » avec openLCA. Comme déjà dit précédemment, on constate que les identifiants originaux des flux n'ont pas été conservés, que ce soit par MEANS-InOut ou openLCA. Pour MEANS-InOut, ceci s'explique par le fait que ecoSpold1 ne supporte pas ces identifiants. De son côté, openLCA re-génère ses propres identifiants. On constate également sur le flux d'exemple que dans MEANS-InOut, le nom est différent, car différent dans SimaPro.

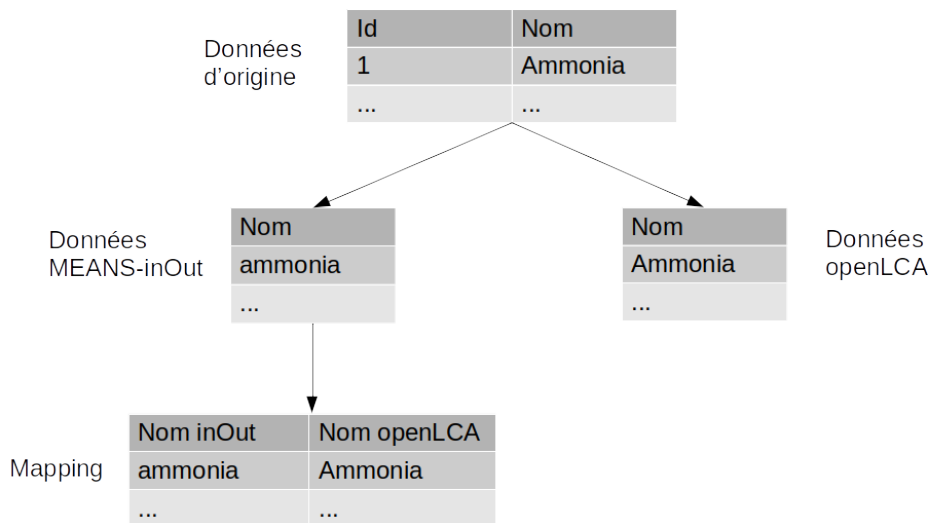


FIGURE 9.3 – Nécessité du mapping entre MEANS-InOut et openLCA

Avant de détailler ce mapping, il est nécessaire de préciser quelques termes.

Définitions

Les définitions suivantes dont la source n'est pas précisée sont issues de la norme ISO 14040 :2006[6], qui décrit les caractéristiques essentielles de l'ACV.

processus Ensemble d'activités interdépendantes ou en interaction qui transforment des intrants (inputs) en extrants (outputs). Par exemple : la production de 20 t de maïs.

échange Entrée ou sortie d'un flux dans un processus². Par exemple : 20 kg d'engrais.

flux Toute entrée ou sortie d'un processus³. Par exemple : du glyphosate, du dioxyde de carbone (CO_2).

flux élémentaire Matériau ou énergie entrant dans le système étudié, issu de l'environnement sans transformation humaine préalable; ou matériau/énergie quittant le système étudié, qui est libéré dans l'environnement sans transformation humaine ultérieure. Par exemple : de l'eau, du CO_2 .

flux produit Produit qui entre ou sort d'un autre système de produits. Par exemple : le glyphosate sort du processus de production du glyphosate, et entre dans le processus de production du maïs.

flux de référence Mesure des sorties d'un processus requise pour remplir la fonction du processus. Par exemple : les 20 t de maïs sont le flux de référence du processus de production de 20 t de maïs. openLCA parle de « quantitative reference ».

Sur la figure 9.4, on retrouve ces termes définis dans le modèle de données (simplifié) d'openLCA. Le type des attributs est renseigné s'il apporte une précision.

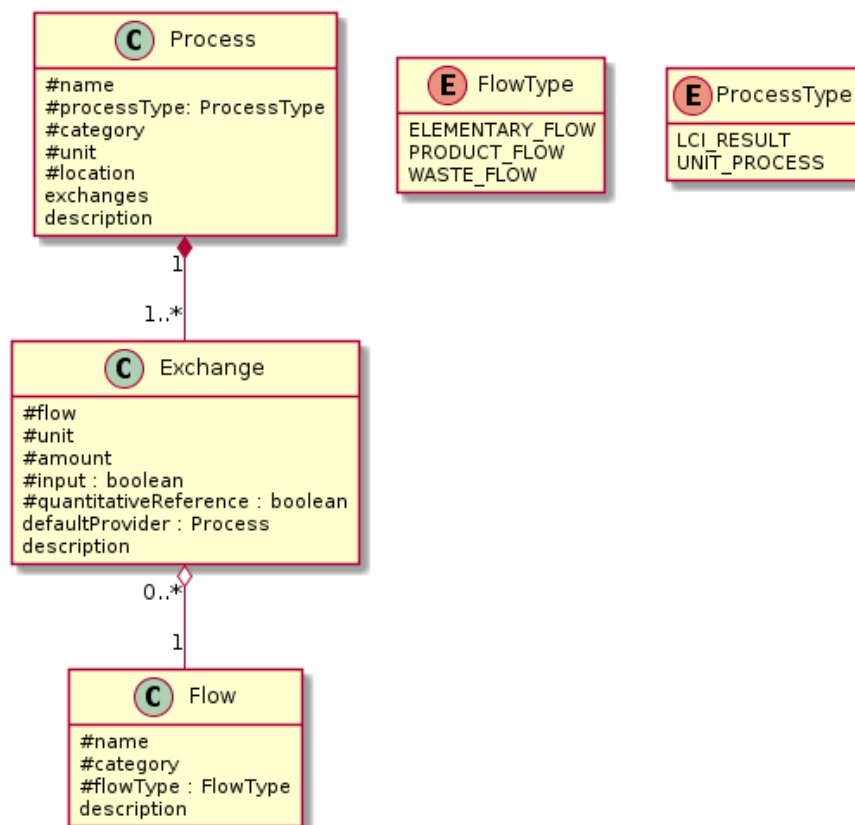


FIGURE 9.4 – UML simplifié du modèle de données d'openLCA

Ces définitions étant posées, il s'agit maintenant de détailler le mapping réalisé.

²<http://greendelta.github.io/olca-schema/Exchange.html>

³<http://greendelta.github.io/olca-schema/Flow.html>

9.2.2 Mapping des flux élémentaires

Un flux élémentaire est identifié par son nom, sa catégorie et son unité. Cette dernière n'a pas posé de problème de mapping.

Incohérence sur les catégories

Les flux élémentaires sont classés dans une hiérarchie à deux niveaux. Il a été constaté que SimaPro et openLCA n'utilisaient pas exactement la même nomenclature. Un mapping a donc d'abord dû être effectué sur les noms des catégories. Le tableau 9.1 en est un extrait (vingt-six catégories au total).

	MEANS-InOut		openLCA	
	Catégorie	Sous-catégorie	Catégorie	Sous-catégorie
1	Air	low. pop., long-term	Emission to air	low population density, long-term
2	Water	groundwater	Emission to water	ground water
3	Soil	agricultural	Emission to soil	agricultural
4	Resource	land	Resource	land
5	Resource	(unspecified)	Resource	unspecified

TABLE 9.1 – Extrait du mapping entre les catégories de flux élémentaires

Ces correspondances ont été construites manuellement, puis rassemblées dans un fichier CSV chargé (une seule fois) par le programme. Il reste alors à effectuer un travail similaire sur les noms des flux.

Incohérence sur les noms

Comme le montre la figure 9.3, les noms des flux élémentaires dans MEANS-InOut et openLCA pouvaient être différents. Bien souvent, le problème était dû à une différence de casse (p. ex. présence d'une majuscule chez openLCA et absence chez SimaPro). Pour des substances chimiques comportant des numéros et des caractères non alphanumériques (p. ex. « 1-Naphthaleneacetic acid »), des différences étaient là aussi souvent constatées.

La stratégie a alors consisté à appliquer une fonction de normalisation au nom fourni par MEANS-InOut, en passant la chaîne de caractères en minuscules, et en retirant tout caractère non alphanumérique (en particulier, les espaces).

Algorithme

La procédure résultante est donnée par l'algorithme suivant :

Code 9.2: Mapping d'un flux élémentaire

Input: un échange e associé à un flux élémentaire

Output: le même échange avec la bonne catégorie et le bon nom, si on l'a trouvé dans la base d'openLCA

Function `mapElementaryFlow(e)` :

```

| category = getOpenLCACategory(e.category);
| name = normalise(e.name);
| targetFlow = olcaDatabase.findFlow(category, name);
| if targetFlow ≠ null then
|   | e.category = targetFlow.category;
|   | e.name = targetFlow.name;
| end
| return

```

9.2.3 Mapping des flux produits

Un flux produit est le résultat d'un processus (cf. section 9.2.1). Il peut être produit par différents processus. Par exemple, un litre de pétrole peut avoir été produit en Arabie Saoudite ou en Russie. Il y a donc bien une distinction entre un produit générique (dans les deux cas, c'est du pétrole), et des processus spécifiques. Dans le schéma de données d'openLCA (cf. figure 9.4), cette distinction se retrouve bien dans la classe `Exchange`, avec une référence vers le `flow` générique, et la référence vers le processus producteur, via l'attribut `defaultProvider`.

Cependant, cette distinction n'a été apportée qu'avec la version 3.4 de la base de données ecoinvent. Auparavant, il n'y avait pas de flux générique, mais des flux spécifiques (un pour le pétrole d'Arabie Saoudite, et un pour celui de Russie). En conséquence, le format `ecoSpold1` (rendu obsolète par la version 3 d'ecoinvent, pour rappel) est basé sur ces flux spécifiques. En réalité, `ecoSpold1` ne fait pas de différence entre flux et processus, puisque le nom de l'échange est en fait le nom du processus. Pour pouvoir retrouver la distinction entre `flow` et `provider`, il est donc nécessaire d'extraire ces informations du nom mis à disposition dans l'`ecoSpold1`. La figure 9.5 montre les deux cas possibles.

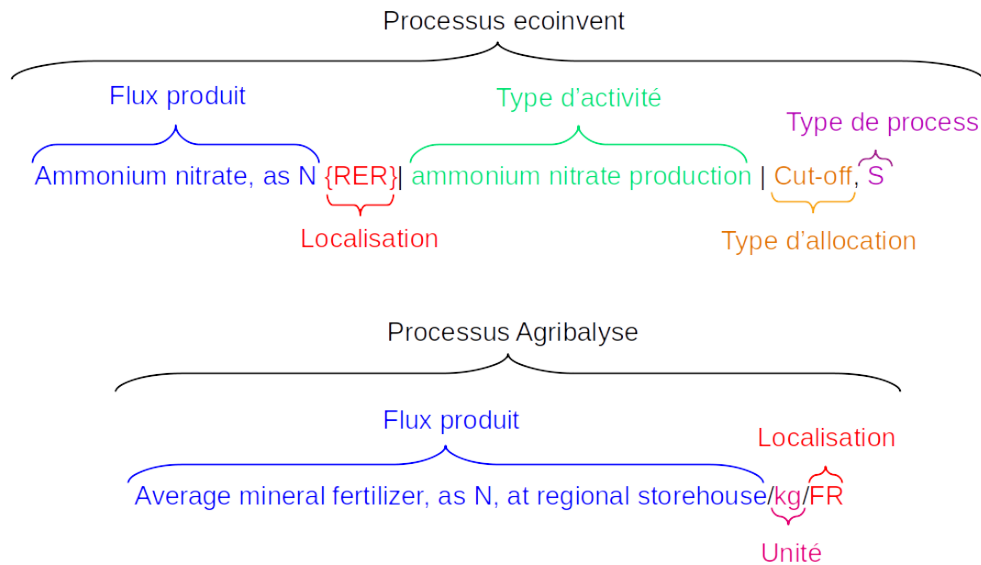


FIGURE 9.5 – Décomposition des noms des processus AGRIBALYSE et ecoinvent formatés pour SimaPro

Pour les processus ecoinvent, par rapport au schéma de données d'openLCA, on note la présence d'un « type d'activité » et d'un « type d'allocation ». Ces deux informations n'ont pas d'attribut propre dans le schéma d'openLCA, mais sont bien présentes dans le nom des processus.

La stratégie a donc consisté, via des expressions régulières, à d'abord déterminer si l'échange courant était issu d'ecoinvent ou d'AGRIBALYSE, puis à trouver le flux produit et provider correspondants dans openLCA. Encore une fois, en l'absence de tout identifiant, la correspondance ne peut se faire que sur les noms des entités. Il y a donc de nouveau eu recours à la fonction de normalisation déjà mentionnée.

L'algorithme ne va pas être détaillé, mais on peut noter que le format ecoSpold1 ne permettant pas de spécifier le provider dans un échange, cette information est stockée temporairement, et valorisée une fois l'import terminé, via l'attribut `defaultProvider` (cf. figure 9.4).

9.2.4 Algorithme général : un mapping « à la volée »

La première stratégie a consisté à effectuer ces opérations de mapping directement lors de l'import des objets `ecoSpold1` :

Code 9.3: Import d'un objet `ecoSpold1` dans `openLCA` avec mapping à la volée

```

Input: un objet ecoSpold1 spold
Output: l'ensemble des datasets (processus) de l'ecoSpold1 ont été importés
dans openLCA
Function importEcoSpold1(spold) :
  for d ∈ spold.getDatasets() do
    for e ∈ d.getExchanges() do
      if e.isElementaryFlow() then
        | mapElementaryFlow(e);
      end
      else
        | mapProductFlow(e);
      end
    end
  openLCA.import(d);
end
return

```

Cette solution s'est avérée fonctionnelle, mais a rapidement montré ses limites. D'une part, les stratégies de mapping ne permettent pas de couvrir tous les cas (plus de mille flux présents dans la base de MEANS-InOut). Certaines correspondances ne peuvent ainsi être établies qu'« à la main », grâce à une expertise métier. D'autre part, cette solution n'est évidemment pas optimale, dans la mesure où l'opération de mapping est réalisée à chaque import. Il a tout de même été décidé d'aller au bout de la procédure, en implémentant le calcul d'impacts désormais possible, avant de revenir sur le mapping.

9.3 Implémentation du calcul d'impacts

9.3.1 Utilisation d'une nouvelle librairie d'`openLCA`

Pour réaliser les calculs d'impacts, le module Maven d'`openLCA` requis se nomme `olca-core`⁴. Ce dernier a donc été ajouté aux dépendances de MEANS-InOut, comme le module d'import (cf. section 9.1.1).

9.3.2 Le calcul

Construction d'un product system

Pour calculer l'impact d'un processus (p. ex. la production de 20 t de maïs), `openLCA` requiert d'abord de créer un product system.

product system Collection de processus avec des flux élémentaires et de produits, qui remplissent une ou plusieurs fonctions définies, et qui modélisent le cycle de vie d'un produit[6]. Par exemple : le processus de production des graines des maïs, d'un engrais, l'approvisionnement en eau, le travail mécanique (et donc, le processus de production

⁴<https://github.com/GreenDelta/olca-modules/tree/master/olca-core>

du tracteur et du carburant utilisés) forment le product system qui modélise le cycle de vie des 20 t de maïs produites.

Ce product system est construit à partir du processus de référence de l'ecoSpold1 importé (cf. balise `referenceFunction` dans le code 8.1). C'est lors de cette construction qu'openLCA fait le lien entre les flux produits et leur provider. Dans le cas de l'import depuis MEANS-InOut, ces liens se font directement en utilisant l'attribut `defaultProvider` des échanges (cf. figure 9.4).

Création d'un cache sur la base d'openLCA

Pour que le calcul fonctionne, il est nécessaire de créer un cache sur la base de données d'openLCA. Cette opération se fait directement via une opération définie par openLCA, et dure environ trente secondes.

Initialement, ce cache était créé à chaque calcul, ce qui ajoutait une durée fixe au temps de calcul, et ne réalisait donc pas du tout l'objectif d'un cache. L'idéal est de créer ce cache au démarrage de l'application. Aucune opération similaire n'était actuellement réalisée dans MEANS-InOut, en revanche certaines opérations étaient définies lors de la connexion d'un utilisateur. La solution temporaire a donc été de créer le cache lors de la connexion du premier utilisateur après le démarrage du serveur. Finalement, cette alternative a pu être supprimée, et le cache est désormais bel et bien créé au démarrage du serveur.

Choix de la méthode de calcul

Il existe de nombreuses méthodes de calcul d'impacts, qui définissent donc des indicateurs d'impact. Par exemple, la méthode ILCD (International Reference Life Cycle Data System), souvent utilisée par MEANS, en définit seize : l'impact sur le changement climatique, sur l'eutrophisation marine, terrestre, la destruction de la couche d'ozone, etc.

Le calcul utilise pour l'instant cette méthode ILCD, car bien connue de l'équipe MEANS, en particulier son agronome, donc un choix pertinent dans l'optique de la comparaison des résultats avec SimaPro. Une possibilité non prioritaire dans le cadre du stage, mais facile à mettre en œuvre serait de proposer différentes méthodes de calcul dans l'interface de MEANS-InOut.

Configuration du calcul

openLCA propose différentes configurations de calcul, de la plus simple qui donne simplement l'impact du processus de référence, à la plus complexe qui détaille l'impact de chaque processus impliqué dans le product system.

Pour une première version, une configuration simple a été choisie, l'objectif étant surtout de pouvoir ensuite comparer les résultats d'impacts du processus de référence avec SimaPro.

L'étude des différentes configurations possibles a été amorcée avec l'agronome de l'équipe MEANS. Les résultats pour un même ICV ont ainsi été générés selon les neuf configurations possibles. L'objectif était ici d'obtenir un fichier de résultats qui apporte les mêmes informations que SimaPro, en particulier concernant la « contribution » des différents processus (p. ex. est-ce l'épandage des engrais ou le travail du sol qui impacte le plus la qualité du sol?) de l'ICV. En consultant la documentation d'openLCA, certaines configurations semblent ne pas avoir de sens, mais l'information n'étant pas assez

explicite, une question a été posée sur le forum dédié, mais sans réponse pour l’instant. À l’issue du stage, la bonne configuration du calcul reste donc à trouver, pour correspondre au mieux au besoin des utilisateurs de MEANS.

Présentation des résultats à l’utilisateur

L’utilisateur reçoit un fichier ZIP téléchargeable contenant le tableau des résultats et un fichier texte d’information sur les processus non pris en compte dans le calcul. Ces processus dits « fantômes » ne contiennent pas d’intrants, car ces derniers n’ont pu être définis via MEANS-InOut. L’utilisateur devra donc les compléter lui-même s’il le souhaite, dans un logiciel d’ACV. Ceci suppose que le ZIP contienne un fichier d’import de l’ICV. Cette fonctionnalité existe pour SimaPro, et peut être développée simplement, en appelant les fonctions déjà définies par openLCA.

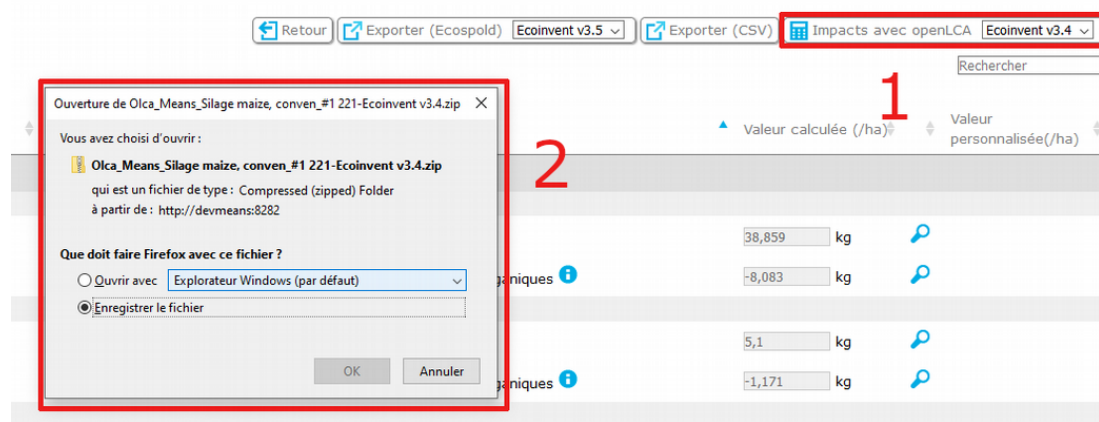


FIGURE 9.6 – Capture d’écran du calcul avec openLCA depuis MEANS-InOut

L’obtention des résultats d’impacts se fait donc désormais uniquement en cliquant sur le bouton de calcul. Il n’est plus nécessaire d’exporter l’ICV en ecoSpold1, d’ouvrir SimaPro, de sélectionner les bases à utiliser, d’importer l’ICV, de sélectionner la méthode de calcul, avant d’avant lancer le calcul à proprement parler. Dans le cas des productions végétales simples (cf. 9.5.1), le calcul n’a pas duré plus de 11 secondes, et pas plus de 21 secondes pour les productions végétales complexes (cf. 9.6).

9.4 Améliorations

Après les premiers tests, les résultats obtenus étaient trop différents de ceux renvoyés par SimaPro. Plusieurs points d’amélioration ont donc été traités, notamment avec l’agronome de MEANS.

9.4.1 Faire le mapping en amont

Premièrement, pour que les résultats soient bons, il faut que l’ensemble des flux soient mappés. Or, comme relevé en section 9.2.4, le mapping à la volée n’était pas satisfaisant, tant en termes d’exhaustivité des flux mappés, que de performance.

Il a donc été décidé que le mapping serait fait en amont, avec un fichier CSV (à terme, son contenu devra être migré dans la base de MEANS-InOut) qui contient les correspondances entre flux. Une fonction dédiée a été développée à cet effet. Elle permet

de mapper une base de flux donnée de MEANS-InOut avec la base correspondante dans openLCA. On obtient un fichier avec les flux mappés et un avec les non mappés. Pour ces derniers, une colonne du CSV indique la raison de l'échec : aucun candidat au mapping, plusieurs candidats, aucun provider, plusieurs providers (pour les flux produits). Si plusieurs candidats ont été trouvés, ils sont renseignés.

Sur les 1277 flux contenus dans la base de MEANS-InOut, 3% n'ont pu être mappés directement par l'algorithme. Le problème se posait surtout pour les substances chimiques, qui peuvent avoir plusieurs noms, et comporter des caractères non alphanumériques. Pour trouver les correspondants de ces flux, l'expertise de l'agronome de l'équipe MEANS a été nécessaire.

Cet algorithme de mapping permet ainsi d'envisager l'intégration des nouvelles versions des bases de données d'ACV. Le mapping de la version 3.5 d'ecoinvent a ainsi pu être réalisé, et celui de la version 3.0 d'AGRIBALYSE (sortie en juin) amorcé. La fonction se lance directement via une URL, il n'a pas été jugé prioritaire de développer une interface étant donné que c'est une fonctionnalité d'administration de MEANS-InOut. Une documentation a été rédigée et une transmission de la maîtrise de la fonctionnalité effectuée avec un membre de l'équipe.

9.4.2 Compenser la différence des méthodes de calcul

Une autre cause des différences de valeurs d'impacts était le fait que la version de la méthode de calcul ILCD disponible sous SimaPro ne l'était pas sous openLCA (la première supérieure à l'autre).

La première conséquence est que certains « facteurs d'impacts » étaient différents. Par exemple, l'impact sur le changement climatique de l'ammoniaque n'aurait pas la même valeur entre les deux versions. Dans ces conditions, difficile d'obtenir les mêmes résultats. L'expertise de l'agronome de MEANS a donc été nécessaire pour identifier les différences de facteurs d'impacts, en partant des différences de résultats. Les valeurs ont alors été modifiées à la main dans la méthode ILCD d'openLCA.

Le deuxième problème était que certains flux pris en compte dans la version SimaPro ne l'étaient pas sous openLCA. Par exemple, l'impact de l'ammoniaque serait pris en compte dans la première, et pas dans l'autre. Or, il a été constaté que si un flux classé dans une sous-catégorie donnée (p. ex. « land » dans la catégorie « Ressource », cf. tableau 9.1) n'était pas pris en compte, le même flux mais dans la sous-catégorie « unspecified » pouvait lui être pris en compte. Une fonction de « post-traitement » qui fait suite à l'import de l'ICV a donc été développée pour couvrir ces cas, et ainsi avoir un maximum de flux qui seront pris en compte par la méthode de calcul.

Ces améliorations faites, la comparaison des résultats pouvait être de nouveau envisagée.

9.5 Comparaison des résultats avec SimaPro

9.5.1 Méthode

Échantillon de test

Pour rappel, l'objectif premier était de faire fonctionner le module de calcul sur des productions végétales simples, par opposition aux successions de cultures, dont le cas a pu être traité par la suite. Un échantillon de trois ICV représentatifs du cas général et des cas particuliers a été sélectionné pour les tests :

Un ensilage⁵ de maïs (12 t) en système conventionnel⁶ Production végétale simple, avec un seul produit en sortie : les 12 t d'ensilage.

Une prairie cultivée en système conventionnel Une production avec cette fois-ci deux produits en sortie : 18 t d'ensilage d'herbe et 16 t d'enrubannage⁷ (on parle ici de co-produit). Un schéma possible pour une prairie est en effet par exemple de l'ensiler en mai, puis de faire un enrubannage une fois que l'herbe a repoussé, vers mi-juin. Le module de calcul doit donc renvoyer l'impact respectif de ces deux productions, donc deux tableurs distincts dans le fichier ZIP.

Du blé (3 t) cultivé en système biologique La particularité de cette production est qu'elle repose sur une « semence de ferme » : les graines utilisées pour le semis (environ 170 kg) proviennent de la production elle-même. Autrement dit, une première culture de blé a été réalisée l'année n , des graines ont été gardées, et utilisées pour le semis d'une culture identique, l'année $n + 1$. Ainsi, la production s'auto-référence dans ses intrants. SimaPro était capable de gérer ce cas particulier, il faut tester si openLCA également.

Critère de comparaison

L'agronome de l'équipe MEANS a identifié quatre catégories d'impact « robustes » aux comparaisons parmi les seize de la méthode ILCD. On mesure donc le taux de différence des résultats renvoyés par SimaPro et openLCA, sur ces quatre catégories. Elle a également fixé que les résultats seraient jugés fiables en-dessous de 5% de différence.

9.5.2 Comparaison

Le tableau 9.2 consigne les différences (en pourcentage) entre les résultats d'impacts obtenus avec SimaPro et openLCA, sur les quatre catégories d'intérêt :

	ensilage de maïs	prairie ensilage	prairie enrubannage	blé
Changement climatique	0	0	0	0
Acidification terre et eau douce	0	0	0	344
Eutrophisation eau douce	1	0	0	0
Eutrophisation eau de mer	0	0	0	0

TABLE 9.2 – Comparaison des résultats d'impacts pour des productions végétales simples

Sauf l'impact de la culture de blé sur l'acidification de la terre et de l'eau douce, le taux de différence ne dépasse pas 1%. Les causes de l'unique valeur aberrante ont commencé à être recherchées avec l'agronome de MEANS, mais ce travail resterait à achever.

⁵Une plante ensilée est conservée dans un silo, sous une bâche étanche, et sera donnée comme fourrage aux animaux.

⁶Par opposition à l'agriculture biologique et à toute autre méthode agricole sans intrant chimique et/ou sans labour des sols.

⁷Une plante enrubannée est d'abord fauchée, puis rassemblée en blocs qui seront mis sous film plastique (« enrubannés »). Également donné en fourrage aux animaux.

L'extension du module de calcul à des productions plus complexes va permettre d'éprouver la robustesse de ses résultats, notamment pour lever le doute quant à cette valeur aberrante.

9.6 Extension aux successions de cultures

Alors que les productions végétales sur lesquelles le module a jusqu'alors été testé ne comportaient qu'une seule culture, il s'agit ici de s'intéresser aux successions de cultures. Par exemple, une parcelle où l'on aurait successivement cultivé du maïs, du blé, du colza puis de l'orge, sur trois ans.

Une légère modification a d'abord dû être apportée à la fonction d'import pour pouvoir gérer plusieurs objets `ecoSpold1`.

Pour vérifier les résultats des calculs, comme pour les productions simples, trois scénarios de cultures ont été choisis. Ces derniers ne seront pas détaillés, mais le nombre de cultures qu'ils contenaient peut être précisé : six pour les deux premiers et cinq pour le dernier. Le module de calcul renvoie donc un tableau de résultats pour chaque culture du scénario, et un pour le scénario en lui-même. Ce sont les valeurs de ce dernier qui ont été comparées, avec 2,25 % de différence dans le pire des cas.

La prochaine étape, non traitée pendant le stage, sera d'étendre le module de calcul aux productions animales. Les ajouts de code sont minimes (principalement rendre le bouton de calcul accessible dans l'interface et créer une route dédiée dans la partie backend de l'application), le principal travail sera la comparaison des résultats. Pour les productions piscicoles, une base de données devra être mappée avec le module dédié.

Conclusion

Ce stage a atteint son objectif principal, à savoir développer un module de calcul d'impacts environnementaux dans MEANS-InOut. Les grandes étapes ayant mené à cette finalité ont été le choix d'openLCA comme outil de calcul, le mapping de ses bases de données avec MEANS-InOut, le développement du module d'import, puis de calcul, et enfin la comparaison des résultats obtenus avec l'ancien logiciel d'ACV, SimaPro. Plusieurs perspectives s'ouvrent alors pour MEANS, principalement le calcul d'impacts en série, des études de sensibilité plus rapides à mettre en œuvre, l'extension de l'utilisation de l'application et la collaboration avec d'autres projets (cf. 3.2.5). Le lien établi avec openLCA ouvre également la possibilité d'utiliser davantage des fonctionnalités de cet outil, en particulier la spatialisation des impacts, et l'intégration de la dimension sociale voire économique (les deux autres dimensions de la durabilité avec l'environnement) dans les indicateurs d'impacts. Plusieurs améliorations sont envisageables. En premier lieu, améliorer la configuration de calcul pour qu'elle corresponde au mieux aux besoins des utilisateurs de MEANS-InOut. Par ailleurs, le module de calcul est appelé à être étendu aux productions animales. Le nécessaire a également été fait pour que le module puisse supporter les futures versions des bases de données d'ACV. Enfin, et même si ce travail apparaît moins prioritaire, il serait sans doute bénéfique de ne plus passer par le format ecoSpold1, mais de directement utiliser le format développé par openLCA, JSON-LD.

Le principal point positif du stage est que MEANS-InOut, et à travers elle MEANS puisse désormais communiquer directement avec un logiciel d'ACV. Le point négatif est le temps passé sur le mapping des bases de données. D'une part, ce problème est lié aux problèmes d'inter-opérabilité inhérents aux bases de données d'ACV, du fait du grand nombre de formats d'échange de données, du respect relatif des standards, et des modifications que les logiciels d'ACV apportent eux-mêmes aux bases originales. D'autre part, le temps passé s'explique par mon manque de connaissances sur l'ACV. D'où une difficulté à anticiper les différents problèmes de formats, et une avancée par essai-erreur. Sans-doute aurais-je dû consacrer davantage de temps à me documenter avant de me lancer dans le développement.

Ce stage a confirmé la réalité de problématiques étudiées dans ma filière Ingénierie des Connaissances et des Supports d'Information, notamment la difficulté à rendre et maintenir des données inter-opérables. La tension réside dans le fait que d'une part, la diversité des acteurs produisant et utilisant ces données est une des principales causes du problème, et d'autre part on ne peut que souhaiter cette diversité, qui rend possible la richesse des projets utilisant ces données. L'ouverture vers le web de données permise par le format JSON-LD, développé par openLCA, est probablement une piste intéressante pour favoriser l'inter-opérabilité des données d'ACV.

Bibliographie

- [1] Anne-Claire ASSELIN-BALENÇON et al. *AGRIBALYSE v3.0 : the French agricultural and food LCI database*. Technical report. ADEME, 2020. URL : <https://ecolab.gitbook.io/documentation-agribalyse/documentation-complete#documentation-relative-a-la-construction-des-donnees-de-la-base-agribalyse-r-3-0>.
- [2] Julie AUBERGER. “La plateforme MEANS : objectifs, méthodes et outils”. In : Ecole-Chercheurs INRA. *Évaluation multicritère pour la gestion des écosystèmes forestiers, prairiaux et aquatiques*. (5 avr. 2019).
- [3] GREENDELTA. *olca-modules*. Version 1.9.0. 26 juin 2020. URL : <https://github.com/GreenDelta/olca-modules>.
- [4] GREENDELTA, éd. *The open source Life Cycle and Sustainability Assesment software*. 2020. URL : www.openlca.org (visité le 26/06/2020).
- [5] Elisabeth GUICHARD. “BaGaTel : an ontology driven database on food composition, transformation process, nutritional and sensory quality”. In : 9e journées de la SFCi. (21 nov. 2019). URL : https://sfci2019.sciencesconf.org/data/pages/SFCi_abstract_Guichard_poster.pdf.
- [6] *ISO 14040 :2006. Environmental management — Life cycle assessment — Principles and framework*. Standard. Genève, ch : International Organization for Standardization, juil. 2006.
- [7] MEANS. *MEANS-InOut un logiciel pour l'écoconception des systèmes agricoles*. MEANS, 2019. URL : <https://www6.inrae.fr/means/Media/Fichier/MEANS/Ecoconception-MEANS-fr-web>.
- [8] Chris MUTEL. *Brightway2 LCA framework*. 2020. URL : <https://brightway.dev> (visité le 26/06/2020).
- [9] Caroline PÉNICAUD. *Présentation du projet DataSusFood*. 2020. URL : https://www6.inrae.fr/datasusfood_eng/Presentation (visité le 17/07/2020).
- [10] Michel SEBILLOTTE. “Agronomie et agriculture : essai d’analyse des tâches de l’agronome”. In : *Cahiers ORSTOM. Série Biologie* (1974), p. 3-25. URL : <http://www.documentation.ird.fr/hor/fdi:17681>.
- [11] Michel SEBILLOTTE. “Itinéraires techniques et évolution de la pensée agronomique”. In : *Comptes-rendus de l’Académie d’Agriculture de France* (1978), p. 906-913.
- [12] Gregor WERNET et al. “The ecoinvent database version 3 (part I) : overview and methodology”. In : *The International Journal of Life Cycle Assessment* 21.9 (2016), p. 1218-1230. DOI : 10.1007/s11367-016-1087-8. URL : <http://link.springer.com/10.1007/s11367-016-1087-8>.

Acronymes

ACV

Analyse du Cycle de Vie. 8–10, 12–14, 16, 19, 21–25, 28, 35, 36, 39, *Glossaire* : ACV

ICV

Inventaire de Cycle de Vie. 9, 12, 14–16, 18, 21–23, 25, 26, 34–36, 42, *Glossaire* : ICV

INRAE

Institut National de Recherche pour l’Agriculture, l’alimentation et l’Environnement. 6, 8

ITK

Itinéraire technique. 9, 12–14, 16, *Glossaire* : ITK

MEANS

MulticritEria AssessmeNt of Sustainability. 7–9, 12–14, 17, 19, 21–24, 34–37, 39

SAS

Sol Agro et hydrosystème Spatialisation. 6–8, 22, 23

UMR

Unité Mixte de Recherche. 6–8, 22, 23

Glossaire

ACV

L'Analyse du Cycle de Vie (ACV) est une méthode d'évaluation normalisée (ISO 1040 - 44) permettant de réaliser un bilan environnemental multicritère quantitatif d'un produit ou d'un service. Cette méthode permet d'avoir une vision globale des impacts environnementaux d'un produit et d'en identifier les origines. 8

ecoSpold

Format d'échange de données pour des ICV développé par ecoinvent. 21, 23–25

flux

Toute entrée ou sortie d'un processus. Par exemple : du glyphosate, du dioxyde de carbone (CO_2). 29

flux de référence

Mesure des sorties d'un processus requise pour remplir la fonction du processus. Par exemple : les 20 t de maïs sont le flux de référence du processus de production de 20 t de maïs. 29

flux produit

Produit qui entre ou sort d'un autre système de produits. Par exemple : le glyphosate sort du processus de production du glyphosate, et entre dans le processus de production du maïs. 29, 31, 32, 34

flux élémentaire

Matériau ou énergie entrant dans le système étudié, issu de l'environnement sans transformation humaine préalable ; ou matériau/énergie quittant le système étudié, qui est libéré dans l'environnement sans transformation humaine ultérieure. Par exemple : de l'eau, du CO_2 . 29–31

ICV

L'Inventaire de Cycle de Vie (ITK) est l'inventaire des intrants (consommations de ressources) et émissions (de polluants) d'un ITK. 9, *Glossaire* : ITK

ITK

L'Itinéraire technique (ITK) est la combinaison logique et ordonnée des techniques mises en œuvre sur une parcelle en vue d'obtenir une production. 9

processus

Ensemble d'activités interdépendantes ou en interaction qui transforment des intrants (inputs) en extrants (outputs). Par exemple : la production de 20 t de maïs. 28, 31, 33

product system

Collection de processus unitaires avec des flux élémentaires et de produits, qui remplissent une ou plusieurs fonctions définies, et qui modélisent le cycle de vie d'un produit. Par exemple : le processus de production des graines des maïs, d'un engrais, l'approvisionnement en eau, le travail mécanique (et donc, le processus de production du tracteur et du carburant utilisés) forment le product system qui modélise le cycle de vie des 20 t de maïs produites. 33, 34

provider

Se dit d'un processus vis-à-vis de son flux de référence. Par exemple : le processus de production d'un litre de pétrole en Arabie Saoudite est un provider du flux « pétrole ». 31, 32, 34, 36

échange

Entrée ou sortie d'un flux dans un processus. Par exemple : 20 kg d'engrais. 29, 31, 32, 34