

Méthodes et outils informatiques du Cati Sicpa

Alexandre Journaux¹, Thierry Heirman¹, Matthieu Reichstadt², Thierry Coudert³, David Robelin¹, Pierre Chalier⁴

Résumé. Les informaticiens du Cati Sicpa utilisent des méthodes et outils communs pour l'analyse, le développement et la gestion de projets informatiques. Ces pratiques uniformisées ont pour but d'apporter une cohérence de nos applications, sécuriser les bases de données et permettre l'interconnexion des systèmes d'informations. Mais aussi, pour les utilisateurs, l'objectif est d'améliorer leur prise en main des applications et faciliter la gestion de l'assistance. Ainsi, les informaticiens du Cati Sicpa ont à leur disposition une valise méthodologique et technique contenant une forge, une charte ergonomique, des langages de programmations, des frameworks, une méthode d'analyse, des serveurs sécurisés...

Mots clés : modélisation, diagramme, projet, spécification, bases de données, sauvegarde, sécurité, PHP, Symfony, C#, framework, interface, harmonisation, interopérabilité, webservices, Forge, suivi de version, ticket d'assistance

Introduction

Dès sa création en 2012, le Cati Sicpa s'est donné comme objectif de créer des collaborations entre les informaticiens des deux Départements GA et Phase (Génétique Animale et Physiologie Animale et Systèmes d'Élevage). Cela s'est d'abord traduit par la mise en place de différents groupes de travail. Ces groupes de travail sont l'occasion d'échanger sur les différentes technologies, de mettre en place des outils communs et d'aboutir à des recommandations de pratiques et de développement. Les différentes origines des membres du Cati ont enrichi les échanges et chacun a pu apporter ses expertises pour aboutir à la mise en place de méthodes et d'outils communs. Grâce à la bonne dynamique de ces groupes, plusieurs actions et résultats ont vu le jour : des formations (UML, Symfony...), des méthodes (charte ergonomique, méthodologie pour la gestion de projets...) et des outils (forge, webservices...).

UML

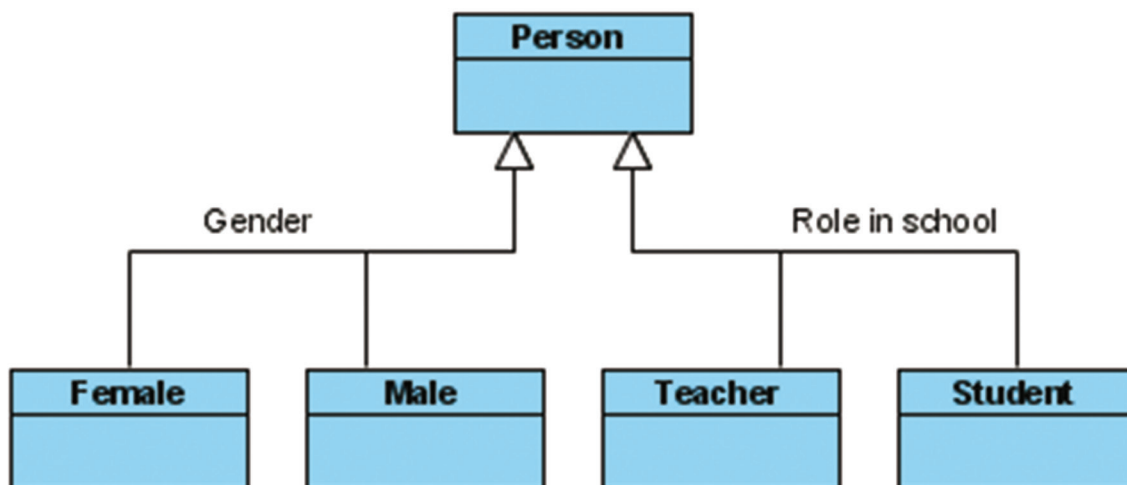
UML (unified modeling language) est un langage de modélisation graphique, utilisé notamment pour la conception d'applications informatiques. UML propose un ensemble de diagrammes, indépendants des langages de programmation, permettant de spécifier l'architecture et le comportement d'un logiciel ou d'un système.

De par son aspect graphique, UML permet de représenter une application informatique de manière compréhensible par des non-informaticiens. C'est un langage qui fournit un standard et des méthodes de conception indispensables dans la phase d'élaboration d'un projet.

Les diagrammes

La version 2.3 d'UML propose 14 types de diagrammes, répartis en diagrammes de structure, de comportement et d'interactions. Le Cati Sicpa a choisi d'utiliser six de ces diagrammes.

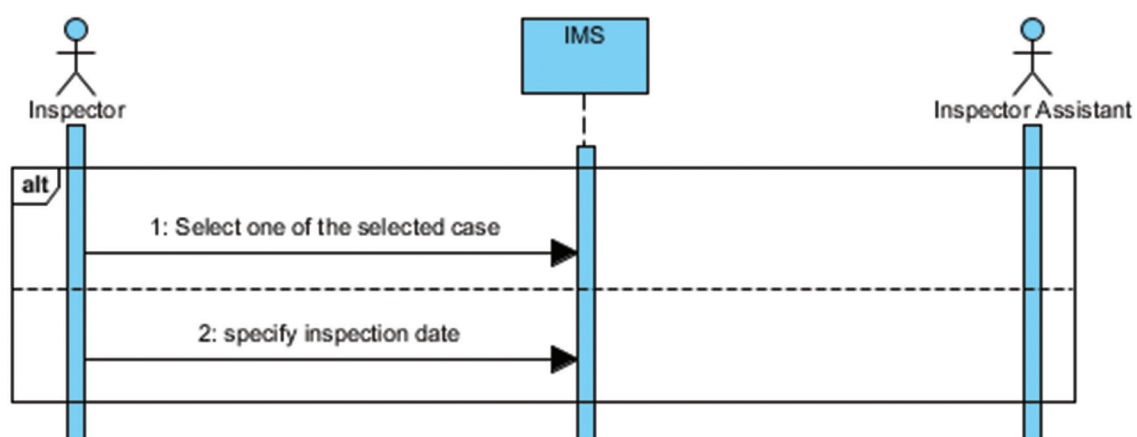
1 UMR GenPhySE, Inra, 31326 Castanet -Tolosan Cedex, France
2 UMR Herbivores, Inra, 63122 Saint-Genès Champanelle, France
3 CTIG, Inra, Domaine de Vilvert, 78352 Jouy-en-Josas Cedex, France
4 UE HerbiPôle, Inra, 63820 Laqueuille, France
alexandre.journaux@inra.fr



Le **diagramme de classes (Figure 1)** est le diagramme central d'UML, le plus utilisé. Il représente la structure du système sous formes de classes (les objets du système) et de relations entre ces classes. Une classe contient des propriétés et des comportements. Ce diagramme permet la génération du schéma de bases de données.



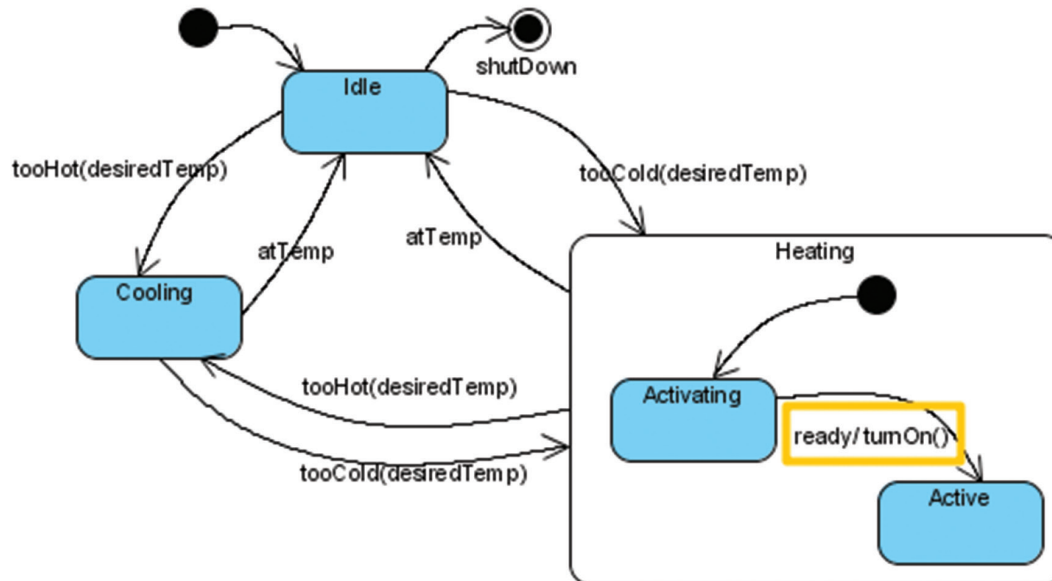
Le **diagramme de cas d'utilisation (Figure 2)** structure les besoins des utilisateurs et les objectifs du système. Il représente l'interaction entre les utilisateurs et le système, sans en modéliser le comportement interne.



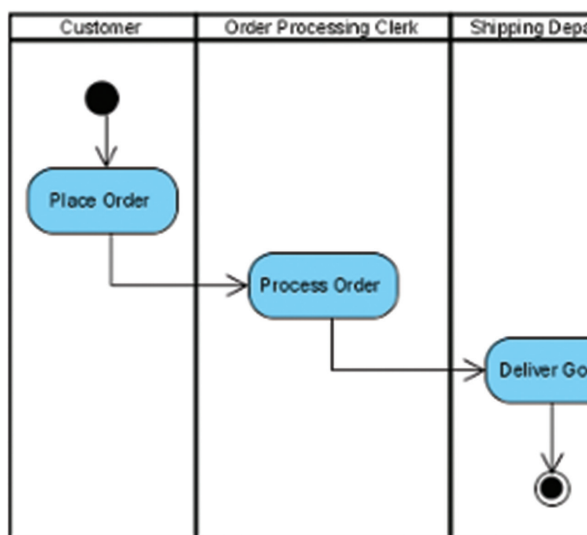
Le **diagramme de séquences (Figure 3)** représente les interactions entre les acteurs et le système de manière chronologique, sous formes d'envoi de messages. Les cas d'utilisation sont ici détaillés en utilisant une ligne de temps.



Le **diagramme de collaboration / communication (Figure 4)** représente de façon simplifiée un diagramme de séquence en se concentrant sur les échanges de messages entre les objets.



Le **diagramme d'états-transitions (Figure 5)** représente le comportement du système sous la forme d'une machine à états : il décrit les séquences possibles d'états et d'actions par lesquelles un objet peut passer tout au long de sa vie.



Le **diagramme d'activités (Figure 6)** représente le comportement du système sous forme de flux ou d'enchaînement d'activités, avec une notion de synchronisation.



Gestion de projet

Du groupe UML vers le groupe gestion de projet

Un groupe de travail « Modélisation UML » est créé dès 2012. Son but est d'améliorer la qualité de la démarche d'analyse en s'appuyant sur le langage UML. Ce groupe choisit un outil de modélisation, monte une formation collective à ce langage puis détermine les diagrammes à utiliser pour l'analyse.

Le groupe décide, ensuite, de mieux formaliser l'analyse en proposant une démarche commune de gestion de projet informatique.

Gestion d'un projet informatique

Après une journée de sensibilisation à la gestion d'un projet informatique, le groupe propose une démarche à suivre pour le démarrage de tous les nouveaux projets du Cati. Cette démarche s'appuie sur une définition précise des acteurs et sur l'élaboration de plusieurs documents : un plan projet, un guide pour les cahiers des charges et un modèle pour les spécifications fonctionnelles.

- ✓ Les acteurs dans le projet sont déterminés dès le lancement du projet. Il convient ainsi de nommer un chef de projet métier et un chef de projet informatique. Les membres des comités utilisateurs et comité de pilotage sont désignés ensuite.
- ✓ Le plan projet est un document qui décrit la façon dont le projet va s'organiser. Il précise notamment la fréquence et le rôle des différentes instances et acteurs.
- ✓ Un guide de cahier des charges est proposé pour préciser ce qui est attendu dans ce document : expression du besoin fonctionnel, identification des acteurs, liste des contraintes...
- ✓ Un modèle de spécifications fonctionnelles qui permet de traduire l'expression des besoins, formulée dans le cahier des charges, en différents cas d'usages détaillés à l'aide des diagrammes UML (diagramme de cas d'utilisation et diagramme de séquence).

Perspective

La démarche commune et l'ensemble des documents proposés par le Cati, offrent désormais un guide de conduite de projet pour les responsables de systèmes d'informations. En outre, cela permet une organisation harmonisée pour le démarrage de tous nouveaux projets informatiques.

Les bases de données

Collaboration du CTIG

Le CTIG (Centre de Traitement de l'Information Génétique) est le centre informatique du Département de Génétique Animale. Une de ses missions est la gestion des infrastructures de calcul et d'hébergement de services informatiques pour le Département GA. C'est dans ce cadre qu'il met à disposition du Cati Sicpa les moyens informatiques nécessaires à son activité : serveurs, stockage, bases de données, sauvegardes...

Les serveurs

DGA5

Ce serveur physique possède les caractéristiques suivantes :

- 64Go de mémoire
- quatre processeurs Power5 (2 core) à 2100 Mhz

La version 10g du SGBD (système de gestion de bases de données) Oracle y est installée. Onze bases de données sont hébergées sur ce serveur. Parmi elles, se trouvent les bases des systèmes d'informations Sicpa Ovins/Caprins, Sicpa Volailles, Sicpa Bovins et Sicpa Porcins.

DGA3

Ce serveur virtuel possède les caractéristiques suivantes :

- 32Go de mémoire
- huit processeurs

La version 10.0.32 du SGBD MariaDB y est installée. Six bases de données sont hébergées sur ce serveur. Parmi elles, se trouvent les bases des systèmes d'informations Sicpa Lapins, Sicpa Sanitaire, Sicpa Alimentation et Sicpa Expérimentations.

Les sauvegardes

Pour toutes ces bases de données, une procédure de sauvegarde est mise en place : backup du filesystem pour Oracle, mysqldump pour MySQL. Dans les deux cas, des tests de restauration ont été réalisés avec succès. En plus de ces sauvegardes un PRA (plan de reprise d'activité) a été défini pour DGA3 avec des tests de restaurations sur un site distant.

L'accès sécurisé aux données

Chaque base est accessible par l'intermédiaire d'un port TCP (transmission control protocol) différent. De plus, l'accès est filtré via des règles sur la zone CTIG des Firewall Inra.

Langage de programmation C#

C# (prononcé « Ci sharp ») est un langage de programmation développé par Microsoft, destiné à être exécuté par la plateforme « .NET ». Il s'agit d'un langage orienté objet, proche du langage Java. Il peut être utilisé pour développer des applications web (dites « clients légers ») ou des applications de bureau (dites « clients lourds »). Le Cati Sicpa développe actuellement des applications de bureau grâce au langage C#, en utilisant la technologie WinForms (ou Windows Forms). Il s'agit de l'interface graphique incluse dans la plateforme .NET, qui permet de réaliser des applications en fenêtres, classiques de l'environnement Windows.

Environnement de développement

Visual Studio est l'environnement de développement intégré (IDE : integrated development environment) utilisé par le Cati Sicpa. Il regroupe au sein d'une même interface tous les outils utiles aux développeurs : création des fenêtres graphiques par sélection de composants, éditeur de code, compilateur, debugger, tests automatisés, outil de contrôle de version pour sauvegarder et partager le code entre plusieurs développeurs.

Frameworks utilisés

Le Cati Sicpa préconise également l'utilisation de l'ORM (object-relational mapping) NHibernate (<http://nhibernate.info/>), framework open source qui gère la persistance des objets en bases de données relationnelles. NHibernate permet, grâce à un mappage entre les objets définis dans le code de l'application et les tables



de la base de données, de simplifier l'enregistrement et l'interrogation des données. En effet, il permet de remplacer les accès à la base de données par l'appel des méthodes définies sur les objets. Afin d'harmoniser les interfaces, les frameworks ObjectListView (<http://objectlistview.sourceforge.net/>) et MetroFramework (<https://thielj.github.io/MetroFramework/>) sont utilisés dans tous les nouveaux développements pour applications de bureau. L'ensemble de ces frameworks est désormais regroupé dans un framework maison, développé en interne, afin de faciliter et homogénéiser les développements C#.

PHP et Symfony

Il existe de nombreuses technologies de développement, notamment les technologies du web. Parmi les langages utilisés afin de créer des applications web, le Cati Sicpa utilise PHP, et basé sur ce langage, le framework Symfony.

Le langage PHP

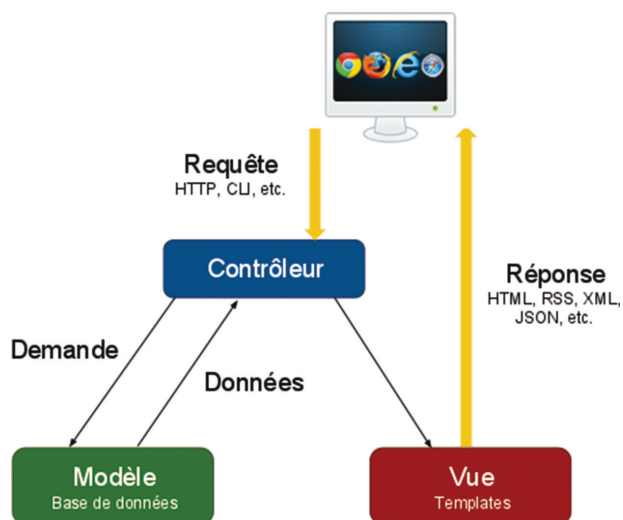
Le langage PHP (hypertext preprocessor) est un langage de programmation open-source utilisé dans les technologies du web. Il a pour rôle de fournir un comportement dynamique aux pages web. C'est un langage qui peut être utilisé de manière procédurale, ou de manière orientée objet.

C'est dans ce dernier cadre qu'ont été mis en place différents frameworks web, dont Symfony qui est utilisé au sein du Cati. Un framework est un outil de travail qui fournit de nombreux composants afin de proposer un cadre pour le développement, dans l'objectif de faciliter celui-ci et de le rendre le plus propre possible.

Le framework Symfony (Figure 7)

Le framework Symfony (actuellement en version 3) utilise la structure MVC (modèle/vue/contrôleur) qui permet de bien séparer les différentes parties du développement que sont la base de données, les interfaces de saisie et de récupération des informations, et le moyen d'accéder à toutes ces informations.

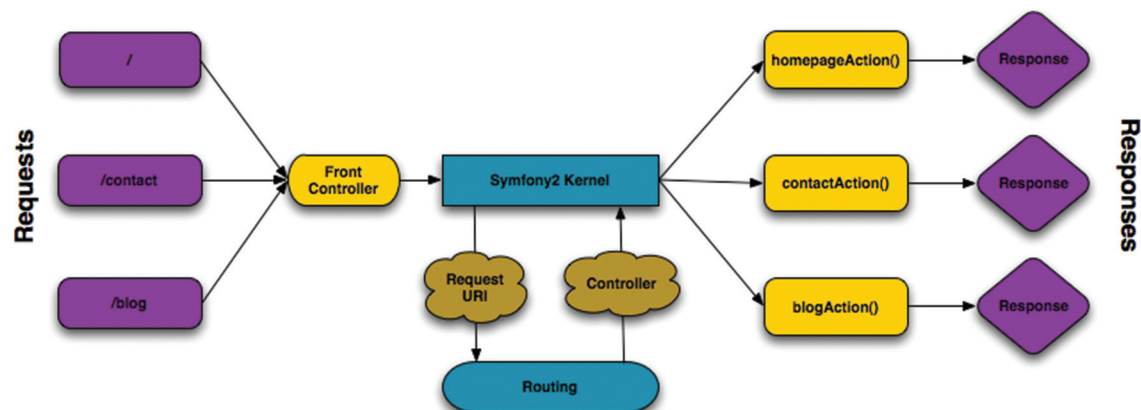
C'est un des frameworks PHP les plus utilisés dans le monde, grâce notamment à la forte mobilisation de la communauté de développeurs qui fournit de nombreux bundles (codes apportant une ou plusieurs fonctionnalités, comme par exemple la gestion des utilisateurs, l'interface administration...). Il a été créé par une société française (SensioLabs).



Symfony est basé sur certains principes :

- ✓ les bundles pour bien séparer les différentes parties des applications qui contiennent
 - les entités, qui représentent grosso modo les tables de notre base de données
 - les contrôleurs, qui permettent de relier les entités et les vues (affichage des pages web)
- ✓ les services qui représentent des fonctionnalités à développer afin de fournir des exécutions systématiques de code (par exemple : les mails, la validation des données, de l'antispam...)
- ✓ la séparation des environnements, avec un environnement de développement qui permet d'améliorer les fonctionnalités des applications, et un environnement de production pour les utilisateurs.

Exemple de diagramme de fonctionnement de Symfony (Figure 8)



L'utilisateur se positionne sur une page définie par une URL spécifique. Cette page est associée à une route (qui fait le lien entre une adresse et un contrôleur). Le cœur de Symfony récupère cette route et exécute la méthode associée à la route, dans le contrôleur correspondant. Cette méthode va renvoyer une réponse, généralement une vue HTML qui va afficher la page sur le navigateur de l'utilisateur.

Une formation en interne à Symfony (disponible en ligne : wiki.inra.fr/wiki/wisards/Symfony/) a été dispensée à tous les développeurs du Cati Sicpa. PHP associé à ce framework est donc préconisé par le Cati pour les nouveaux développements Web (ex : Sicpa Parcelle, « applications abeilles »...).

La charte ergonomique

En 2014, le Cati Sicpa fait le constat que les applications développées par les membres du Cati sont hétérogènes, non seulement par les langages de programmation utilisés, mais aussi par leurs interfaces. Ceci s'explique par des développements spécifiques, pour chaque espèce, mis en place à des moments différents. Suite à la demande d'utilisateurs et dans l'objectif de mettre en cohérence nos systèmes d'informations, un groupe de travail est créé pour réfléchir à l'harmonisation des interfaces de nos applications. L'arrivée de nouveaux développeurs ainsi que le démarrage de nouveaux projets sont, alors, des opportunités pour la création de ce groupe.

Objectif du groupe de travail

Le groupe de travail se donne un triple objectif :


- ✓ harmonisation des interfaces pour toutes les nouvelles applications développées au sein du Cati ;
- ✓ création d'une charte ergonomique qui décrit les éléments à développer dans les applications afin de garantir une même cohérence d'interface ;
- ✓ création d'un « framework maison » qui contient des bibliothèques communes permettant des développements rapides et cohérents.

Contenu de la charte

L'harmonisation des interfaces offre un double avantage :

- pour les développeurs, cela permet un partage des solutions techniques
- pour les utilisateurs, une prise en main plus rapide d'une application à l'autre.





La charte graphique décrit donc les éléments d'interface à uniformiser. Ce document est composé d'un tronc commun qui donne les recommandations générales et de quatre annexes qui particularisent ces recommandations en fonction du type d'application (client/serveur, mobile PDA (personal digital assistant), web, web mobile).

Perspective

La charte graphique est désormais un document d'entrée pour tout nouveau projet au même titre que le cahier des charges. Elle est appliquée sur tous les nouveaux développements du Cati. Les utilisateurs ont ainsi un apprentissage facilité des applications. Pour les développeurs du Cati, les solutions techniques communes mises en place aboutissent à la création de bibliothèques réutilisables regroupées dans le « framework maison ».

Les outils d'interopérabilité : les webservice et le bus PetalsESB

Constat

Nous avons fait le constat, courant 2010, que certains de nos systèmes d'informations (SI) se complétaient fonctionnellement, mais ils pouvaient demeurer isolés les uns des autres de par leur conception. Le projet GAniMed a donc été lancé afin d'étudier et de mettre en place une solution permettant de faire communiquer les SI entre eux. Aujourd'hui, la solution est utilisée de manière complètement transparente par les utilisateurs des SI que le Cati Sicpa a mis en place.

Interopérabilité des SI dans le Cati Sicpa

L'interopérabilité est la capacité que possède un SI à fonctionner avec d'autres SI, existants ou futurs. Nous avons dû faire trois choix majeurs au cours du projet. Le choix de la méthode au fil de l'eau (c'est-à-dire à la demande de l'utilisateur) s'est imposé rapidement par le besoin de réactivité dans la consultation des données entre nos SI. Ensuite, la technologie choisie est celle de l'*Enterprise Service Bus* (ESB), une technologie qui permet de faire communiquer des applications qui n'ont pas été conçues pour fonctionner ensemble. Enfin, parmi les différents ESB du marché, nous avons choisi *PetalsESB* car il s'agit d'un produit français, nous permettant d'avoir des relations plus faciles avec le personnel support du produit.

Mais la mise en place de l'ESB n'est pas suffisante pour assurer l'interopérabilité entre SI. Il nous manque encore les briques les plus fines : les webservice (**Figure 9a**). Un *webservice* est un programme informatique de taille modeste qui permet de réaliser une action très précise de manière répétée en ne faisant varier que ses paramètres en entrée. Le webservice va agir comme un connecteur entre le bus (ESB) et le SI afin de permettre l'interopérabilité (**Figure 9b**).

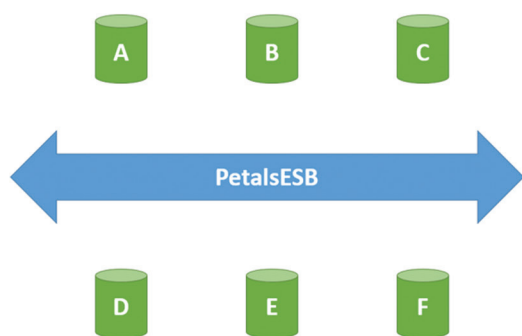


Figure 9a. ESB sans les webservice.

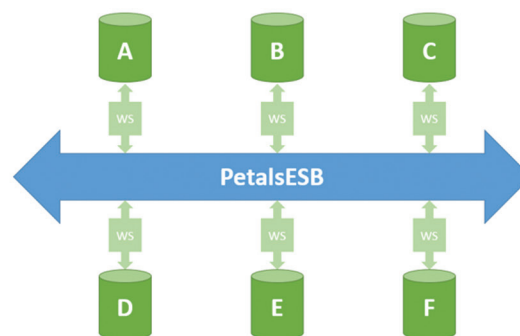


Figure 9b. ESB avec les webservice.

Dès lors, un SI qui expose un webservice sur le bus PetalsESB peut partager certaines de ses fonctionnalités, celles définies dans le webservice, avec tous les autres SI disponibles sur le bus. Imaginons que le webservice publie des données du SI échantillon eSIToul Barcode, tous les autres SI peuvent désormais récupérer leurs données propres enrichies des données concernant les échantillons prélevés sur les animaux.

La technologie SOAP (simple object access protocol) et le langage de programmation Java ont été retenus pour l'implémentation des webservices : SOAP pour satisfaire les prérequis de PetalsESB et Java, car il permet des développements de webservices fiables et rapides grâce aux annotations de son interface de programmation JAX-WS 2.0.

Conclusion

À travers les différents éléments mis en place grâce à ce projet, nous avons pu faire communiquer entre eux des systèmes d'informations qui n'avaient pas été architecturés dans ce but. Cette interopérabilité entre SI permet de croiser des données provenant de 11 SI pour en tirer une plus-value scientifique et éviter les doubles saisies, sans que l'utilisateur final n'ait la moindre action supplémentaire à réaliser. Aujourd'hui, une étude est en cours pour permettre aux scientifiques d'utiliser les webservices directement depuis SAS (statistical analysis system) ou R.

La Forge DGA

La décennie des années 2000 a vu à l'Inra, la popularisation des outils de gestion de version de code source. Rapidement, ces solutions qui se limitent au code, mais peu à son processus d'élaboration et de maintenance, en facilitant l'écriture à plusieurs développeurs, ont popularisé le besoin de disposer d'un environnement plus complet lié à l'ensemble de la gestion de la vie d'un logiciel : la Forge logicielle. De telles solutions ont été installées par des équipes dans des laboratoires pour répondre à ce besoin, et c'est en 2010, sous l'égide du Département de Génétique Animale, que s'est formalisée la mise en place d'une seule forge logicielle dédiée à l'ensemble des membres de ce Département et à leurs collaborateurs : la Forge DGA. Ouverte le premier juillet 2011, elle est accessible à l'adresse suivante : <http://forge-dga.jouy.inra.fr/>.

Organisation

La Forge DGA est hébergée au CTIG qui en assure l'administration système (sauvegarde quotidienne, mises à jour logicielles, évolutions matérielles...). L'administration web (support utilisateur, documentation, dépannage, communication...) est assurée par le Cati Sicpa.





Choix de la solution

Les contraintes retenues pour le choix de la solution à héberger ont été, d'une part, la robustesse et la facilité de maintenance car il n'y avait pas de forces nouvelles affectées à ce projet, et, d'autre part, une ergonomie la plus simple possible afin d'être appréhendée par tout type d'utilisateurs du Département de Génétique Animale, sans sacrifier aux fonctionnalités offertes. Le choix s'est ainsi porté sur Redmine (<http://www.redmine.org/>) associé aux gestionnaires de version Subversion et Git. De plus, la grande communauté utilisatrice de Redmine assure un développement actif du produit.

Fonctionnalités pour les utilisateurs

Presque toutes les fonctionnalités de Redmine sont présentes sur la Forge DGA ; en plus des fonctionnalités classiquement présentes sur une Forge logicielle (gestionnaire de version Git et Subversion, wiki, gestion de ticket ou demandes, système de notifications par email et flux RSS (really simple syndication)), citons entre autres quelques possibilités : présenter un projet publiquement ou en réserver l'accès aux membres déclarés du projet ; avoir plusieurs dépôts par projet ; personnaliser la présentation des tickets/demandes, fonctionner par sous-projets avec une vision globale au niveau du projet principal, fermer un ticket/demande via un message de commit ; lier les tickets/demandes entre eux ; pour le « manager » du projet, activer ou non pour chacun des projets les différentes fonctionnalités ; exporter une page au format pdf (portable document format) (tickets/demandes, wiki).

Utilisateurs et utilisations

D'abord pensées pour répondre aux besoins des développeurs de logiciels et de leurs utilisateurs, les fonctionnalités offertes par la Forge DGA répondent également à d'autres besoins par les solutions collaboratives et de gestion de projet qu'elle propose. À gros traits, quatre types d'utilisation de la Forge DGA se sont dessinés. 1. Développement de logiciels internes à l'Inra. 2. Développement de logiciels dédiés à une problématique scientifique précise. 3. Espace de travail individuel, partagé ou non. 4. Gestion communautaire non-informatique. Remarque : une Forge reste avant tout un outil opérationnel pensé pour l'organisation et la réalisation d'un travail ; ce n'est pas un outil adapté à la communication externe, comme pourrait l'être un site web par exemple, bien que certaines fonctionnalités pourraient le laisser croire.

Quelques chiffres (septembre 2017)

1380 utilisateurs enregistrés ; 535 projets hébergés.

Conclusion

La mise en place de méthodes et d'outils communs ont d'abord l'avantage, côté informaticiens, de faciliter la gestion et le développement d'un nouveau système d'informations. En effet, la mise à disposition de méthodes, de langages, de frameworks et d'outils communs facilite la prise en main d'un nouveau projet et favorise l'entraide entre les développeurs.

Tout ce travail offre aussi, pour les utilisateurs, de nombreux avantages : aide pour l'expression du besoin, assistance standardisée, même ergonomie et donc apprentissage plus rapide des applications...

Cette démarche de partage et d'uniformisation des méthodes et des outils va continuer car elle a ainsi montré son utilité et son efficacité.