


<p>ECOLE CENTRALE D'ELECTRONIQUE  GROUPE ECE</p>	<p>RAPPORT DE STAGE 2^{ème} année du cycle ingénieur</p>
<p><u>ELEVE-INGENIEUR :</u></p> <p>Nom : CAILLARD Prénom : Clément</p>	<p><u>MAJEURE :</u></p> <p><input type="checkbox"/> Systèmes d'Information & Réseaux <input type="checkbox"/> Systèmes Embarqués <input type="checkbox"/> Télécommunications & Réseaux</p>
<p><u>ENTREPRISE :</u></p> <p>Nom : Institut de Recherche et Coordination Acoustique/Musique (IRCAM) Adresse : 1, place Igor Stravinsky, 75004 Paris Adresse du lieu de stage si différent : Confidentiel : <input type="checkbox"/> oui <input type="checkbox"/> non Signature du Maître de Stage :</p>	
<p><u>DESCRIPTION DE LA MISSION :</u></p> <p>Stage de développement : base de données et environnement de publication Internet, en environnement Linux, Apache, MySQL, PHP (LAMP).</p> <p>L'objet du stage porte sur le développement des formulaires de saisie, des procédures d'interrogation de la base de données, de la mise en forme des échanges de données, de la réalisation de divers composants du système, et enfin des procédures de test et de validation.</p> <p>Compétences requises: PHP, SQL (MySQL), XML (XForms, XSL...)</p> <p>Nombre de pages : 27 pages</p>	

REMERCIEMENTS

Je tiens à remercier tous ceux qui, de près ou de loin, m'ont permis de réaliser mon stage et d'écrire ce rapport ; ainsi que les personnes qui m'ont prêté des documents ou qui m'ont apporté leurs témoignages à l'occasion de rencontres, notamment :

Jérôme Barthélemy, mon maître de stage, Responsable du service en ligne, pour ses directives,

*Nicolas Donin, responsable du service analyse des pratiques musicales, pour son accueil,
Samuel Goldszmidt, responsable du développement du projet, pour sa disponibilité, son aide et ses conseils,*

Jacques Theureau, chercheur au CNRS, pour sa bonne humeur et ses anecdotes,

Benoît Luce, stagiaire au service analyse des pratiques musicales,

Mathieu Blardone, stagiaire au service en ligne.

Table des matières

INTRODUCTION	5
--------------	---

PRESENTATION DE L'ENTREPRISE

1. DESCRIPTION DE L'ENTREPRISE	6
1.1 – ORIGINES, CONTEXTE ET STATUT	6
1.2 – ACTIVITÉS	6
2. DESCRIPTION DU SERVICE	7

PRESENTATION DE LA MISSION

1. CONTEXTE, UTILITÉ	9
2. OBJET DE LA MISSION	10
2.1 – OBJET	10
2.2 – COMPÉTENCES REQUISES	10
3. CAHIER DES CHARGES	10
3.1 – CAHIER DES CHARGES	10
3.2 – EXIGENCES	10
4. DÉCOUPAGE DU TRAVAIL DE STAGE	11
4.1 – ACTIVITÉS	11
4.2 – PLANNING PRÉVISIONNEL	12

ASPECTS TECHNIQUES

1. PRÉSENTATION DU SYSTÈME	13
1.1 – DÉFINITION DES OBJETS	13
1.1.1 – Les ressources (Ressource)	13
1.1.2 – Les compositeurs (Person)	14
1.1.3 – Les oeuvres (Work)	14
1.1.4 – Les documents (Document)	15
1.2 – LA BASE DE DONNÉES MYSQL	15
1.2.1 – Le principe	15
1.2.2 – Les objets	17
1.2.3 – Remarques	18
2. ETUDE DES EXIGENCES ET RECHERCHES DE DOCUMENTATION	18
2.1 – FACILITER LA MAINTENANCE	18
2.1.1 – Exigences	18
2.1.2 – Choix de conception	19
2.2 – MINIMISER L'UTILISATION DU JAVASCRIPT	19
2.3 – UTILISER UN ENCODAGE PERFORMANT	20
2.3.1 – Exigences	20
2.3.2 – Choix de conception	20
2.3.3 – Problèmes d'encodage	20

<u>2.4 – VALIDITÉ XHTML</u>	21
<u>2.4.1 – Exigences</u>	21
<u>2.4.2 – Choix de conception</u>	22
<u>2.5 – FORMULAIRES D'ADMINISTRATION</u>	22
<u>2.5.1 – Exigences</u>	22
<u>2.5.2 – XForms : une technologie pertinente</u>	22
<u>2.5.3 – Veille technologique XForms</u>	23
<u>2.5.4 – Choix de conception</u>	25
<u>2.5.5 – Difficultés rencontrées avec la technologie XForms</u>	25
<u>3. DÉVELOPPEMENT DE LA PARTIE NAVIGATION</u>	25
<u>3.1 – PRINCIPE</u>	25
<u>3.2 – CONCEPTION DES INTERFACES</u>	26
<u>3.2.1 – Interface avec la base de données</u>	26
<u>3.2.2 – IHM de navigation</u>	27
<u>3.2.3 – Schéma conceptuel</u>	27
<u>3.3 – SPÉCIFICATIONS SYSTÈME</u>	28
<u>3.4 – DÉCOUPAGE EN FICHIERS</u>	29
<u>3.5 – PRINCIPALES DIFFICULTÉS</u>	29
<u>3.5.1 – Erreurs dans la base de données</u>	29
<u>3.5.2 – Problème du référencement</u>	30
<u>3.5.3 – Outil de recherche</u>	30
<u>3.5.4 – Problèmes de présentation</u>	30
<u>4. DÉVELOPPEMENT DE LA PARTIE ADMINISTRATION</u>	31
<u>4.1 – PRINCIPE</u>	31
<u>4.2 – CONCEPTION DES INTERFACES</u>	31
<u>4.2.1 – IHM d'identification</u>	31
<u>4.2.2 – IHM d'administration</u>	32
<u>4.2.3 – Interface entre le système et les formulaires XForms</u>	32
<u>4.2.4 – Interface interne aux XForms</u>	33
<u>4.2.5 – Schéma conceptuel</u>	34
<u>4.3 – SPÉCIFICATIONS SYSTÈME</u>	34
<u>4.4 – DÉCOUPAGE EN FICHIERS</u>	34
<u>4.5 – PRINCIPALES DIFFICULTÉS</u>	35
<u>4.5.1 – Gestion des relations et publication de données</u>	35
<u>4.5.2 – Problèmes avec XForms</u>	35
<u>4.5.3 – Problèmes d'affichage des formulaires XForms</u>	35

BILAN

<u>1. ETAT DU PROJET</u>	36
<u>2. SUITES ET AMÉLIORATIONS POSSIBLES</u>	36
<u>3. APPORTS POUR L'ENTREPRISE</u>	37
<u>4. INTÉRÊT PERSONNEL</u>	37
BIBLIOGRAPHIE	37
ANNEXES	38

Introduction

J'ai effectué mon stage « technique » du 24 avril au 4 août 2006 au sein du département Recherche et Développement, à l'Institut de Recherche et Coordination Acoustique/Musique (IRCAM), situé à Paris en face du centre George Pompidou. L'Ircam est le plus gros centre de recherche scientifique au monde en ce qui concerne les technologies pour la création musicale.

L'objet de mon stage consistait en la refonte de la base « Brahms », une encyclopédie multimédia sur la musique contemporaine. En effet, 10 ans après sa création, des difficultés conséquentes d'administration et de maintenance de cette base sont apparues. En Décembre 2005, il a donc été décidé de procéder à une migration vers un nouveau système technique.

J'étais très désireux de découvrir le fonctionnement d'un centre de recherche. Ce stage, en outre, m'offrait l'opportunité d'approfondir et d'appliquer mes connaissances dans les technologies telles que XML, XSL et CSS que j'avais découvertes durant mon année d'étude, et d'utiliser XForms, une technologie très récente qui sera beaucoup utilisée dans le futur, et dont la pratique me permettait d'enrichir mes compétences.

Dans un premier temps, je présenterai l'Ircam et l'équipe « Services en ligne » dans laquelle j'ai effectué ce stage. Après avoir précisé le contexte et l'objet de ma mission, j'indiquerai le cahier des charges ainsi que le planning qu'on m'avait fixé. Ensuite, je détaillerai les aspects techniques du projet, le principe du système, mes choix de conception ainsi que les difficultés que j'ai rencontrées. Pour finir, j'exposerai le bilan de ce que cette expérience m'a apporté, tant sur le plan personnel que pour ma future vie professionnelle.

Des informations supplémentaires sur l'institut et sur ma mission seront également disponibles en annexe.

Présentation de l'entreprise

1. Description de l'entreprise

1.1 – Origines, contexte et statut

Georges Pompidou initie en 1969 la création de l'Institut de Recherche et Coordination Acoustique/Musique (IRCAM) dont il confie la direction au compositeur et chef d'orchestre Pierre Boulez.

L'Ircam est associé au Centre Pompidou et placé sous la tutelle du ministère de la Culture et de la Communication. Depuis 1995, l'Ircam et le CNRS sont partenaires dans le cadre d'une unité mixte de recherche STMS (Sciences et Technologies de la Musique et du Son). L'Institut dispose d'un statut d'association à but non lucratif reconnue d'utilité publique.

Institut de Recherche et Coordination Acoustique/Musique (IRCAM)

- Adresse : 1, place Igor-Stravinsky, 75004 Paris
- Métro : Hôtel de Ville, Rambuteau, Châtelet, Les Halles
- Tél. : 01 44 78 48 43 | 01 44 78 15 45
- Fax : 01 44 78 15 40
- Web : <http://www.ircam.fr>

1.2 – Activités

L'Ircam, le plus grand centre de recherche scientifique au monde entièrement dédié aux technologies pour la création musicale, accueille une importante population de chercheurs. Il compte à ce jour près de 90 scientifiques.

Outre la recherche, l'Ircam s'occupe de création et de diffusion musicale. Ainsi, depuis 1977, l'Ircam a passé commande d'environ 500 créations. Ce sont ainsi 240 compositeurs qui ont utilisé les résultats de la recherche pour produire leurs oeuvres.

De plus, l'Ircam dispense deux formations universitaires qui le rapprochent du monde de la recherche scientifique : un Mastère Sciences et Technologies et une Formation Interdisciplinaire de Recherche. L'Ircam propose aussi un Cours d'un an traitant de composition et d'informatique musicale. Enfin, plusieurs formations professionnelles sont organisées sur la composition et l'informatique musicale, les logiciels Ircam ou diverses thématiques.

Une Médiathèque très largement informatisée met à disposition des étudiants et des chercheurs, sur accréditation, un important fonds documentaire scientifique et musical.

Enfin, l'Ircam propose un catalogue d'ouvrages et de disques concernant la création musicale contemporaine. Les éditions de l'Ircam se situent en complément des publications proposées par les chercheurs auprès des revues scientifiques internationales.

2. Description du service

Mon stage s'est déroulé au sein du département Recherche et Développement (direction : Hugues Vinet), dans l'équipe « Services en ligne », dirigée par Jérôme Barthélemy.

Ce service est chargé, dans le cadre de projets nationaux et européens, de la conception et du développement de systèmes informatiques en ligne, proposant en particulier un accès à des bases de données sonores et musicales.

En premier lieu, l'équipe assure la mise en place de solutions technologiques opérationnelles permettant la gestion des contenus sonores dans des bases de données, et leur distribution sur Internet ou en Intranet. L'équipe se positionne aussi au sein des projets en tant qu'intégrateur des différents modules développés par les équipes de recherche.

Pour réaliser cet objectif, l'équipe se base sur l'étude et la mise en oeuvre des standards ouverts, aussi bien ceux issus du Web et du World Wide Web Consortium (W3C), que ceux issus de l'ISO et notamment du groupe MPEG.

Cette double activité, orientée à la fois vers le développement et vers l'intégration, s'inscrit dans un environnement en constante évolution. L'adoption de méthodes itératives, essentiellement fondées sur la veille technologique et l'intégration de technologies récentes, permet d'éviter que les systèmes soient dépassés au moment même de leur finalisation.

Cependant, le responsable du développement informatique du projet, Samuel Goldzmidt, travaille en parallèle avec le service « Analyse des pratiques musicales », dirigé par Nicolas Donin, qui mène des recherches en sciences humaines sur les œuvres et les pratiques musicales modernes et contemporaines. Pour cette raison, mon bureau se trouvait au sein de cette équipe.

NB : la plupart de ces informations ont été extraites du site Internet de l'Ircam : www.ircam.fr

Présentation de la mission

Tous les détails de la mission étaient présentés dans le rapport « Analyse fonctionnelle et technique » laissé par Bertrand Chéret, l'ancien responsable de la mission. La plupart des informations ci-dessous sont donc tirées de ce rapport ou m'ont été apportées par mon maître de stage Jérôme Barthélemy.

1. Contexte, utilité

La base « Brahms » (Base Relationnelle d'Articles Hypertextes sur la Musique du XX^e Siècle) est une base de données sur les compositeurs et les œuvres musicales du XX^e et XXI^e siècles. Cette base publiée sur Internet (<http://brahms.ircam.fr>) est une référence pour de nombreux utilisateurs.

Elle a été construite en utilisant un SGBDR (Système de Gestion de Bases de Données Relationnel) propriétaire, le système 4D de la société ACI. La technologie propriétaire présentait plusieurs inconvénients :

- un coût intrinsèque pour l'acquisition du système (licences, mises à jour...),
- un coût en temps de travail pour effectuer la migration nécessaire vers un nouveau système,
- un ensemble de technologies propriétaires, ce qui rend difficile, voire impossible, un certain nombre d'opérations standard dans le domaine des bases de données,
- une compétence à acquérir et à entretenir pour une seule et unique base,
- un contrôle et une manipulation partielle des données,
- une interopérabilité à sens unique (format ouvert vers format propriétaire), a contrario des choix technologiques actuels de l'Ircam.

Le portage nécessaire sous Mac OS X (nouvelle version du système d'exploitation sur Macintosh) a fait apparaître des problèmes de durabilité et de viabilité du système alors que la base continuait d'être très bien référencée par les moteurs de recherche.

De plus, la base souffrait de plusieurs problèmes spécifiques :

- présence dans la base d'éléments superflus,
- incompatibilités avec les mises à jour,
- problèmes de cohérence dans la structure de données (absence de relations par exemple),
- documentation technique quasi inexistante.

Cette situation, combinée avec la nécessité d'une évolution des contenus, imposait une intervention technique. En Décembre 2005, il a alors été décidé de procéder à une refonte de la base, et au remplacement des technologies sous jacentes, ceci incluant la migration de la base vers une base mySQL, et un environnement de publication Internet LAMP (Linux-Apache-mySQL-PHP).

2. Objet de la mission

2.1 – *Objet*

La réalisation de ce projet avait été commencée par Bertrand Chéret et le stage consistait par conséquent à poursuivre son travail. L'objet du stage portait sur le développement des formulaires de saisie, des procédures d'interrogation de la base de données, de la mise en forme des échanges de données, de la réalisation de divers composants du système, et enfin des procédures de test et de validation.

2.2 – *Compétences requises*

Il s'agissait donc d'un stage de développement web sous environnement LAMP requérant des compétences en HTML, XHTML, PHP, SQL, XML, XSL, Xforms, CSS et Javascript (toutes les technologies utilisées sont détaillées en annexe 2).

3. Cahier des charges

3.1 – *Cahier des charges*

Après avoir lu le rapport laissé par Bertrand Cheret, mon maître de stage, Jérôme Barthélemy, le responsable du projet, Samuel Goldszmidt, et moi-même, sommes convenus d'un cahier des charges en plusieurs points :

- la structure du site en général : interface, navigation, référencement, recherche.
- une partie administration : formulaires de saisie et de modification, système de publication et de validation.
- quelques ajouts supplémentaires possibles : outil de statistiques, gestion des droit d'auteur pour les images et le son, recherche experte.

3.2 – *Exigences*

Bien entendu, quelques contraintes de développement m'étaient imposées :

- élaborer un système compatible avec les architectures de type LAMP donc utilisation de PHP 5.1, MySQL 4.1, apache 2...,
- simplifier la maintenance des fonctionnalités,
- faciliter la mise à jour et la maintenance des données et de l'interface (séparation du contenu et de la mise en forme)
- séparer la structure des formulaires d'administration de sa mise en forme et de la technologie employée pour l'interface,
- vérifier la compatibilité avec le format XHTML afin d'être supporté par tous les navigateurs,
- ne pas utiliser trop de Javascript pour éviter les problèmes de compatibilité,

- utiliser un encodage permettant de coder tous les caractères spéciaux et accentués dans toutes les langues,

4. Découpage du travail de stage

4.1 – Activités

Le travail effectué peut se diviser en 5 activités principales :

- **Etude de documents** liés au projet et à son contexte
 - Analyse fonctionnelle et technique
 - Cahier des charges et spécifications générales
- **Etudes des exigences et recherche de documentation**
 - Simplification de la maintenance (programmation objet, XML, XSL, CSS)
 - Etude de compatibilité (W3C, XHTML, encodage UTF8)
 - Veille sur la technologie Xforms (installation de plugin, recherche de documentation, test de formulaires)
- **Développement de la partie navigation**
 - Etude et mise à jour de la base de données
 - Recherche de documentations sur les technologies employées (PHP, XML, XSLT, CSS)
 - Etude du problème de référencement
 - Développement d'un outil de recherche
- **Développement, test et validation de la partie administration**
 - Recherche de documentations sur les technologies employées (XForms)
 - Mise en place d'un système d'identification
 - Conception et développement d'un outil de gestion de contenu de type CMS (procédure de validation, système de retour, upload de données)
 - Test et validation (test sur un serveur local)
- **Travail de fin de stage**
 - Rédaction de documents (notice d'utilisation et manuel technique)
 - Passation du développement (explication du code source et du principe de fonctionnement à la personne en charge de continuer le projet)

L'étude des documents de projet représentait une étape préliminaire indispensable pour parvenir à une bonne compréhension du contexte, du système en général et des problématiques d'installation en particulier. Une fois les exigences définies, il a été possible de chercher des technologies existantes répondant aux spécifications. J'ai alors été amené à proposer un exemple utilisant XForms.

Avant de commencer le développement du site, j'ai effectué une recherche approfondie et un apprentissage des différents langages et bibliothèques à utiliser. J'ai alors pu commencer le développement de la partie navigation avec un outil de recherche, puis le développement de la partie administration avec un outil de gestion de contenu.

Après quoi, la rédaction de la notice d'utilisation, du manuel technique, les derniers tests du site et les tests d'administration ont pu être réalisés.

4.2 – *Planning prévisionnel*

Le planning de mon stage était très linéaire et prévoyait une durée de sécurité pour régler les éventuels problèmes qui ne manqueraient pas de surgir et pour répondre aux exigences apportées lors des réunions de travail.

Le planning a été conçu en fonction de la première réunion de travail fixée au 23 mai 2006. Pour cette date, il fallait que le site soit fonctionnel pour pouvoir montrer l'avancement du projet au directeur Frank Madlener et obtenir son aval pour la suite.

Etude des documents et de la structure de la base. Veille technologique à propos de Xforms.	2 semaines 24/04/06 - 05/05/06
Développement de la partie navigation du site : <ul style="list-style-type: none"> • outil de recherche, • résolution du problème de référencement, • réalisation de 2 feuilles de style CSS (permettant d'expliquer l'utilité de la séparation contenu/contenant) 	2-3 semaines 09/05/06 – 22/05/06
Développement d'un exemple de formulaire Xforms pour la réunion.	

Le 23 mai eut lieu la première réunion de travail programmée avec mon maître de stage Jérôme Barthélemy, le responsable du développement Samuel Goldzmidt, le directeur de l'Ircam Frank Madlener, le responsable de la base Cyril Beros, l'ancienne rédactrice de la base Delphine Oster et enfin Claire Marquet qui s'occupe de la gestion des droits à l'Ircam. A l'issue de cette réunion, il a été décidé de mettre en place un serveur interne à l'Ircam pour que chacun puisse tester le site et émettre des propositions.

Développement de la partie administration du site : <ul style="list-style-type: none"> • système d'identification, • formulaires de saisie et de modification, • mise en place d'une procédure de validation des formulaires accompagnée d'un système de retour, • système d'enregistrement dans la base de données, 	4-5 semaines 24/05/06 - 25/06/06
Développement d'un outil de statistiques. Gestion des droits audio et visuel. Finalisation du site (retour d'exigences, résolution des problèmes éventuels)	5 semaines 26/06/06 – 28/07/06
Rédaction d'un manuel de présentation du site et d'utilisation de la partie administration. Passation du développement.	1 semaine 31/07/06 – 04/08/06

Aspects Techniques

1. Présentation du système

La base « Brahms » répertorie des compositeurs, des œuvres ainsi que des documents comme des biographies ou des notices. Toutes ces ressources sont enregistrées dans une base de données SQL. Le principe du site est de traiter les informations de la base afin de les sortir au format XML puis de les afficher.

1.1 – Définition des objets

Il y a 3 objets principaux : les Compositeurs, les Œuvres et les Documents. Afin d'offrir un modèle cohérent, structuré sur une base identique et extensible à la base de données, les 3 objets sont regroupés sous un seul appelé Ressource. Ainsi chaque Ressource peut être identifiée selon un attribut `objet` dans le schéma XML ou selon un champ `type` dans la base de données.

1.1.1 – Les ressources (Ressource)

Chaque ressource possède son propre format XML qui suit cependant le même schéma global. Les fichiers XML des ressources sont donc divisés en plusieurs parties :

- une partie information générale (propre à chaque objet) de la ressource : elle expose les propriétés de chaque objet (nom, prénom titre, ...),
- une partie administrative : elle expose les données pour l'administration (date de mise à jour, validité de l'objet, informations propres au système, ...),
- une partie relation : elle énumère les relations établies entre l'objet et les autres ressources sous une forme « résumée » (id + label).

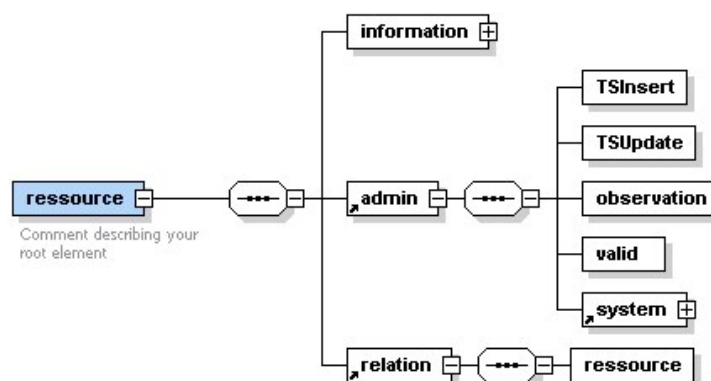


Figure 1 : schéma XML d'une ressource (Ressource)

1.1.2 – Les compositeurs (Person)

Un compositeur est avant tout une personne. Sa description est assez simple, il s'agit des informations principales le concernant. La structure XML d'un compositeur est donc une sorte de carte d'identité détaillant prénom, nom, naissance, décès, nationalité(s) et sexe. Notons qu'il est possible d'ajouter autant de nationalités que nous souhaitons.

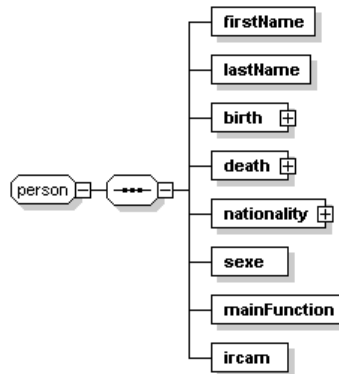


Figure 2 : schéma XML d'un compositeur (Person)

1.1.3 – Les oeuvres (Work)

La partie information d'une œuvre se divise en 4 parties :

- une partie information générale : titre, sous titre, dédicace, ...,
- une partie analyse (`analysis`) : elle expose les caractéristiques de l'œuvre (tonalité, durée, ...)
- une partie relative à l'exécution de sa création (`creationInformation`) : date de création, assistant, studio, ...,
- une partie édition (`publishingInformation`) : lieu d'édition, copyright...

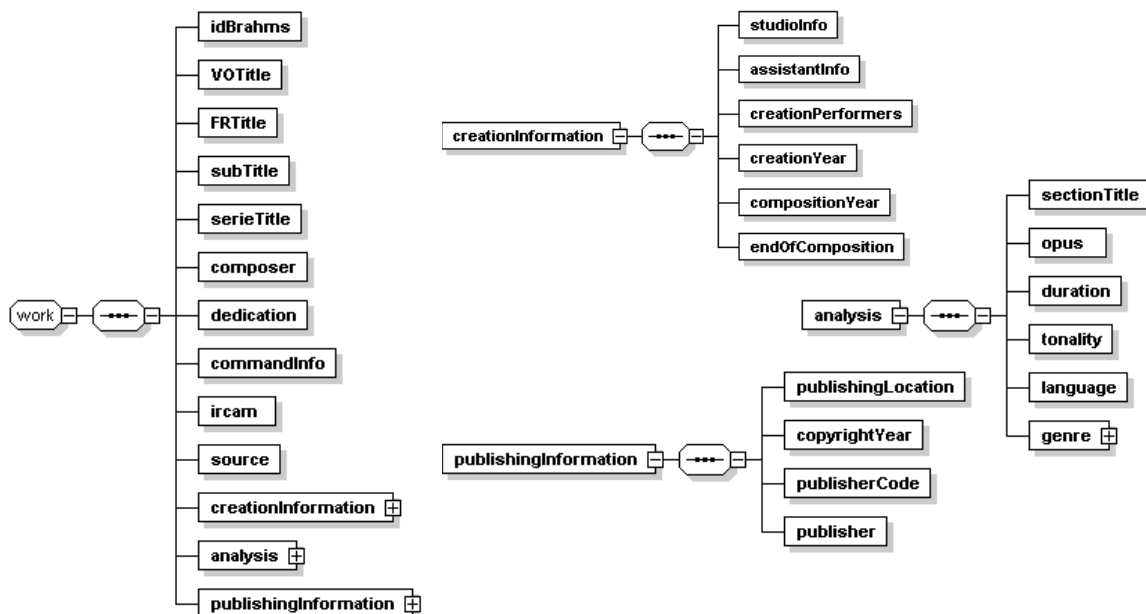


Figure 3 : schémas XML d'une oeuvre (Work)

1.1.4 – Les documents (Document)

Un document se compose d'une partie information générale dont dépend une autre partie se trouvant dans le sous-ensemble `system` du nœud `admin` de la Ressource. En effet il existe deux types de documents : les textes et les images. Enfin, un document contient une dernière partie supplémentaire aux autres objets contenant le texte ou un lien vers l'image.

Pour éviter de surcharger la base de données SQL, il a été décidé de stocker les textes sous forme de fichier XML « en dur » dans le répertoire `/data/text`, les images étant enregistrées dans le répertoire `/data/picture`.

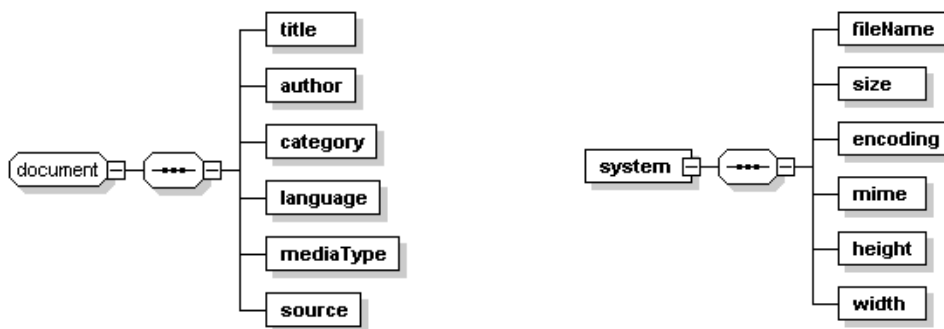


Figure 4 : schémas XML d'un document (Document)

1.2 – La base de données *mysql*

1.2.1 – Le principe

L'architecture de la base de données est assez complexe. Elle est constituée de 20 tables en interclassement `utf8_general_ci` d'une taille totale d'environ 20Mo pour approximativement 110 000 enregistrements.

Le noyau de la base s'articule autour de 5 tables, trois étant relatives à nos trois objets : Compositeur (`person`), Œuvre (`work`), Document (`document`) plus une table (`relation`) intégrant l'ensemble des relations entre ces même objets. Enfin une table centrale (`ressource`) recense l'ensemble des objets sous la forme de ressources identifiées par une clef unique (`id`), un type (`person`, `work` ou `document`) et la validité de l'objet (`valid`). Une ressource valide est publiée sur le site tandis qu'une ressource non valide sera masquée aux visiteurs.

Cette table garantit l'unicité de la ressource et nous permet de centraliser l'administration de la base. Effectivement grâce au moteur InnoDB, qui supporte les contraintes de clés étrangères (`FOREIGN KEY`), on peut administrer la base sans intervenir sur les tables sous-jacentes en cas de suppression ou de mise à jour de la ressource.

Ce mécanisme est très pratique puisqu'il force l'utilisateur à remplir les tables dans un certain ordre lors d'un ajout, qu'il garantit qu'aucun enregistrement « orphelin » ne restera dans la base après une suppression et que la mise à jour n'affectera pas les relations.

Une relation est interprétée, ici, comme l'association de deux ressources, la nature de ce lien étant définie par un prédicat. Il existe sept prédicats :

- notice : entre un document de type notice et une œuvre,
- biography : entre un document de type biographie et un compositeur,
- article : entre un document de type article et une œuvre,
- composer : entre un compositeur et une œuvre,
- livret : entre un document de type livret et une œuvre,
- photography : entre un document de type image et un compositeur,
- reference : entre une référence audio et une œuvre.

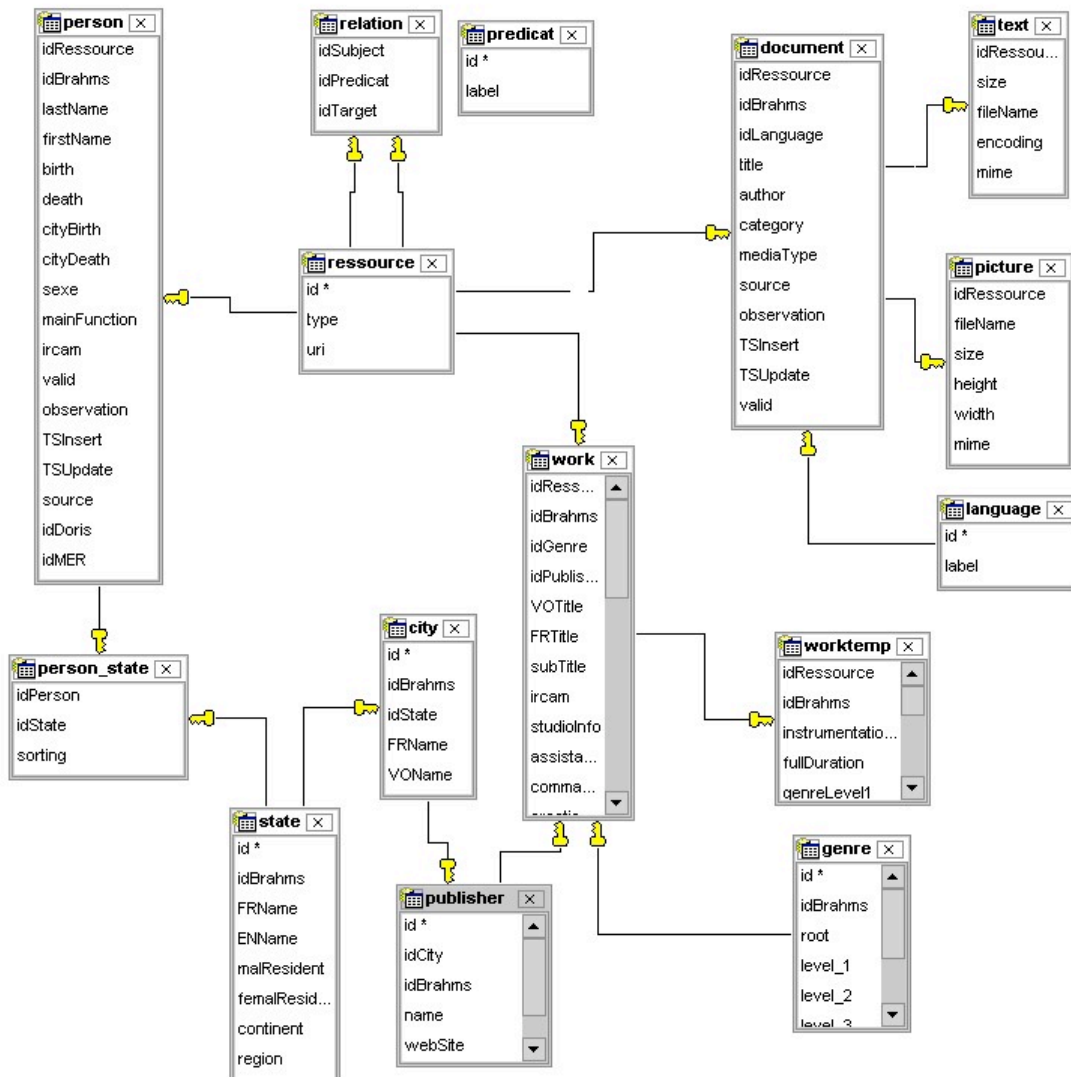


Figure 5 : base de données mysql

1.2.2 – Les objets

- **Les compositeurs (Person)**

L'objet Person met en relations 3 tables :

- la table `person` : elle contient les informations du compositeur,
- la table `state` : elle répertorie tous les pays avec leur localité,
- la table `person_state` : elle associe les deux tables précédentes, permettant ainsi d'associer plusieurs pays à un même individu et de définir sa ou ses nationalités.

- **Les œuvres (Work)**

L'objet Work est lui aussi composé de 3 tables. La table principale `work`, qui contient toutes les informations sur l'œuvre, est liée aux tables `genre` et `publisher`. La table `publisher` répertorie tous les éditeurs. La table `genre` est une copie exacte de la table d'origine. Elle n'est pas optimisée du fait d'un point de vue très particulier dans la réalisation et la définition de la notion de « genre » à l'origine du projet.

- **Les documents (Document)**

L'objet Document se décompose en 3 tables. La table `document` est commune à tous les types de document. Elle contient les informations générales sur le document et le type de document. Selon le type, des données dites système (nom de fichier, taille, largeur, hauteur...) sont contenues dans les tables `text` ou `picture`. Ces tables permettent de retrouver l'image ou le texte du document qui est stocké sur le serveur.

Cette structure est évolutive puisqu'il est possible d'intégrer de nouveaux types de documents en ajoutant une table associée. Par exemple il est envisageable d'ajouter une table `audio` pour les documents de type audio.

Enfin, certains documents `texte` contiennent des liens HTML vers d'autres documents de la base. Pour pouvoir accéder depuis un texte aux documents dans lequel il est lui-même cité, tous les liens entre les documents sont enregistrés dans une table `document_link`.

1.2.3 – Remarques

Le moteur InnoDB a permis de centraliser l'administration. Cela apporte beaucoup d'avantages et une facilité de mise à jour. Mais justement, il faut faire très attention avant de supprimer une ressource : l'erreur n'est pas permise. InnoDB dispose cependant d'un système de récupération automatique en cas de crash de la base.

La structure de la base de données et le modèle des objets ont été repris pour la programmation objet en PHP. Ainsi nous disposons de 3 classes `Person`, `Work` et `Document` héritant d'une classe `Ressource`. Cette construction des classes permet de suivre le système de relation.

L'ensemble du système est donc cohérent tant au niveau XML, SQL que PHP. Le modèle est très détaillé et optimisé mais la gestion des relations, et notamment l'utilisation de la table `relation` que l'on pourrait qualifier « d'intermédiaire », ajoute une complexité au niveau de la construction des requêtes SQL.

2. Etude des exigences et recherches de documentation

2.1 – Faciliter la maintenance

2.1.1 – Exigences

Le système devait impérativement être facile à mettre à jour et à améliorer. En effet, dans un contexte de développement technologique en constante évolution, il faut savoir s'adapter et réagir rapidement pour garder le système opérationnel. Pour simplifier cela, deux choses sont importantes :

- L'utilisation de la programmation objet qui permet de séparer les différentes fonctionnalités en plusieurs modules. Ainsi il est facile de mettre à jour l'un des modules sans modifier les autres et il est aussi possible d'ajouter simplement un nouveau module.
- La séparation du contenu et de la présentation qui permet de changer facilement la mise en forme sans toucher au contenu et inversement.

2.1.2 – Choix de conception

Le langage de programmation utilisé devait pouvoir communiquer facilement avec la base de données SQL, comporter une API pour manipuler des fichiers XML et permettre la programmation orientée objet.

Le choix de PHP version 5 s'imposait, car cette version introduit une programmation orientée objet plus efficace et plus complète, une gestion des erreurs basée sur le modèle des exceptions, ainsi que des fonctionnalités de gestion pour les entreprises ; mais elle apporte surtout des moyens de manipuler des fichiers et des structures XML, basés sur *libXML2* tel l'API SimpleXML utilisé pour le site. Le principal atout de PHP est de pouvoir communiquer avec des systèmes de gestion de bases de données. Il a aussi l'avantage d'être libre, gratuit et simple d'utilisation et d'installation.

Le contenu du site devait être sous forme de fichier XML. Pour le présenter, l'utilisation des feuilles de style XSL (eXtensible Stylesheet Language) était évidente. En effet, XSL est le langage de description de feuilles de style du W3C associé à XML.

Une feuille de style XSL est un fichier qui décrit comment doivent être présentés (c'est-à-dire affichés, imprimés, épelés) les documents XML basés sur une même DTD ou un même schéma. Cette solution nécessite un processeur XSLT permettant de transformer le document XML en un code XHTML.

Enfin une feuille de style CSS permet de présenter le code XHTML de manière à simplifier la navigation (meilleur référencement) et l'accessibilité des contenus (pour les non-voyants par exemple).

2.2 – Minimiser l'utilisation du Javascript

JavaScript est un langage de programmation de type script, orienté objets à prototype, principalement utilisé dans les pages Web. Généralement, JavaScript sert à contrôler les données saisies dans des formulaires HTML, ou à interagir avec le document HTML via l'interface DOM, fournie par le navigateur (on parle alors parfois d'HTML dynamique).

Bien que très utile, Javascript est sujet à quelques incompatibilités avec certains navigateurs, et certains firewall bloquent même son utilisation. Les scripts JavaScript sont souvent la source de difficultés. Le plus souvent, elles sont dues non à des problèmes de portabilité du langage, mais au support des différentes versions des modèles d'objets (DOM) fournis par les navigateurs.

A cause de tous ces problèmes de compatibilité, Javascript ne doit être utilisé que comme une solution de dernier recours ou pour des fonctionnalités obsolètes.

2.3 – Utiliser un encodage performant

2.3.1 – Exigences

La base « Brahms », répertoriant des compositeurs de tous les pays, nécessitait un codage permettant de coder tous les caractères spéciaux et accentués ainsi que les lettres d'alphabet étrangers.

2.3.2 – Choix de conception

Il existe bien évidemment plusieurs codages répondant à ce critère et il fallait donc faire un choix. Comme le système manipulait des fichiers XML, il était plus simple de garder l'encodage par défaut qui correspondait justement aux exigences : l'encodage UTF-8.

UTF-8 (*UCS transformation format 8 bits*) est un format de codage de caractères défini pour les caractères Unicode (UCS) qui intègre l'ASCII et beaucoup de caractères utilisés de par le monde. Chaque caractère est codé sur une suite de un à quatre octets.

Ce codage, contrairement à d'autres comme la série ISO 8859, permet donc de représenter des milliers de caractères de toutes sortes de langues, dont l'ensemble des caractères spécifiques français.

2.3.3 – Problèmes d'encodage

Il se trouve que beaucoup de problèmes d'encodage sont survenus durant le développement. Tout d'abord la base de données n'avait pas été exportée en UTF-8 à l'origine. Les caractères étaient alors codés sur un seul octet et on pouvait donc rencontrer des caractères mal encodés.

Par exemple on pouvait trouver « Ã© » à la place de « é » et ceci pour tous les caractères spéciaux et les caractères accentués.

Il fallait donc convertir l'ensemble des données de la base en UTF-8. J'ai donc changé le type d'encodage de la base de données, des tables et des champs en utilisant l'interface phpMyAdmin. Malheureusement, cela n'a modifié que le type d'encodage sans convertir les données déjà enregistrées.

Après quelques recherches, j'ai pu constater qu'il n'existe aucun système de conversion d'encodage des données en MySQL et j'ai donc pensé à utiliser une fonction PHP qui récupérait le contenu, le codait en UTF-8 puis mettait à jour la table. Mais, bien que parcourant l'ensemble des tables, la fonction refusait de mettre à jour certains champs. Ce dysfonctionnement n'avait aucune logique.

J'ai finalement opté pour une méthode manuelle et radicale : le remplacement de tous les caractères spéciaux, directement en mode texte dans le fichier d'export de la base de données, simplement en utilisant la fonction « rechercher/remplacer » d'un éditeur de texte.

Un autre problème d'encodage concernait la manipulation de fichier XML par des fonctions PHP ainsi que la communication de ces mêmes fonctions avec la base SQL. En effet, pour ajouter des informations dans un fichier XML, il fallait tout d'abord encoder la chaîne de caractères à insérer à l'aide de la fonction `utf8_encode()`.

De plus pour communiquer avec la base de données MySQL, PHP devait indiquer à chaque connexion que le système attendait des résultats utilisant l'encodage UTF-8 grâce aux requêtes : `SET character_set_results = 'UTF8'` et `SET character_set_connection = 'UTF8'`.

J'ai, enfin, constaté que certains fichiers XML correspondant aux informations textuelles des documents étaient mal formés. La plupart étaient tout simplement vides et n'avaient donc aucune raison d'exister. Pour les autres, il s'agissait d'un problème d'encodage ; il a alors fallu remplacer les caractères spéciaux et accentués « à la main » en mode texte. D'autres étaient réellement malformés avec des problèmes de balisage à régler au cas par cas.

2.4 – Validité XHTML

2.4.1 – Exigences

Afin d'être compatible avec tous les navigateurs, mais aussi pour suivre les recommandations du W3C, le site devait être entièrement valide XHTML.

XHTML est un langage balisé servant à l'écriture de pages du World Wide Web. XHTML est le successeur de HTML (HyperText Markup Language), XHTML respectant la syntaxe définie par XML, plus récente et plus simple que la syntaxe définie par SGML respectée par HTML.

XHTML demande une certaine rigueur dans son écriture. Ainsi pour vérifier la compatibilité XHTML d'un document il faut veiller à plusieurs points :

- Le doctype qui définit le type du document est obligatoire et doit être placé tout en haut de la page,
- Fermer correctement les balises et dans le bon ordre,

- Les balises ne possédant pas d'éléments de fermeture doivent être fermées à l'aide d'un espace suivi d'un slash (
, , ...). L'espace n'est pas obligatoire mais permet une compatibilité accrue avec les anciens navigateurs.
- Respecter la casse : en XHTML toutes les balises et les noms d'attributs doivent être écrits en minuscules.
- Toutes les valeurs des attributs doivent être encadrées par des guillemets doubles. Tous les attributs doivent avoir une valeur explicite.
- Les URL contenant le caractère & doivent suivre une écriture correcte et être remplacées par leurs entités HTML, ainsi le '&' devient '&'.
- L'annonce de javascript doit se faire de cette manière : <script type="text/javascript">.

2.4.2 – Choix de conception

Le doctype définit le type du document XHTML. Il est désormais obligatoire et doit être placé tout en haut de la page XHTML. Le doctype à privilégier est celui de l'XHTML strict qui tolère moins les aléas de codage d'un site Web, et il permet donc de coder plus rigoureusement. Il existe cependant trois doctypes :

- Le doctype XHTML Strict, celui à utiliser de préférence (qui ne tolère plus les balises dépréciées en XHTML comme <u>, <i>...),
- Le doctype XHTML Transitional, qui tolère les balises dépréciées en XHTML comme <u>, <i>...,,
- Le doctype XHTML Frameset, qui permet l'utilisation des cadres.

Le site n'étant qu'au stade de développement, il n'y avait aucune transition à effectuer et j'ai donc choisi le doctype strict. (<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">)

J'ai donc vérifié tout le code source grâce à l'outil de validation disponible sur le site W3C : XHTML validator (<http://validator.w3.org/>).

2.5 – Formulaires d'administration

2.5.1 – Exigences

Le système étant compatible XHTML, il y avait la volonté d'utiliser des formulaires plus performants que les formulaires HTML classiques. L'objectif était de séparer la structure des formulaires d'administration de sa mise en forme et de la technologie employée pour l'interface.

Il fallait donc trouver une technologie conforme aux attentes et on m'a conseillé de me tourner vers la récente technologie XForms.

2.5.2 – XForms : une technologie pertinente

XForms est un dialecte XML servant à créer des formulaires en ligne destinés à être utilisés avec HTML, XHTML, WML ou SVG. XForms est une spécification du W3C.

Xforms possède plusieurs avantages par rapport au formulaire HTML classique :

- Séparation de la forme, du contenu et de la logique : cela permet d'utiliser le même formulaire dans un navigateur de micro-ordinateur ou sur un téléphone portable.
- Réutilisabilité : le formulaire est indépendant de la page qui le contient.
- Typage fort : XForms emploie un typage fort (ROFL), ce qui veut dire qu'on peut déterminer finement les données acceptables par le formulaire.
- Internationalisation : l'interface peut exister en plusieurs langues.

Utilisant des documents XML, la technologie Xforms permettait de sortir un formulaire sur le même format que les documents XML extraits de la base. Ce mécanisme permet de faciliter la mise à jour des données.

Cette technologie était donc très bien adaptée à la contrainte posée mais aussi à l'ensemble du système.

2.5.3 – Veille technologique XForms

Un formulaire XForms est divisé en 3 sections :

- Données (*XForms Data Model*): les données qui doivent être lues par le formulaire, et qui seront envoyées sous forme de flux XML au serveur,
- Interface utilisateur : la manière de présenter le formulaire. Cette partie doit être spécifique pour chaque type de support,
- Traitement : les traitements à effectuer.

- **Le modèle de données (*XForms Data Model*)**

Le modèle de données correspond au format d'entrée et de sortie du formulaire. Il s'écrit donc comme un fichier XML placé entre des balises `<model>` puis `<instance>` dans le head de la page. Il est possible de placer plusieurs instances, identifiées grâce à l'attribut `id`, dans un même modèle. Il peut aussi exister plusieurs modèles. Après les instances, on pose les conditions de soumission et les contraintes spéciales.

```
<xforms:model id="model1">
  <xforms:instance id="instance1">
    <person>
      <fname>prénom</fname>
      <lname />
    </person>
  </xforms:instance>
  <xforms:submission id="form1" action="submit.php" method="post" />
</xforms:model>
```

Comme nous pouvons le constater dans l'exemple précédent, il est possible de donner des valeurs d'origine à l'instance. Dans ce cas, le champ pour saisir l'information correspondante à ce nœud XML est automatiquement pré-rempli.

Il existe plusieurs méthodes de sortie en XForms que l'on peut choisir à l'aide de l'attribut `method` de l'élément `submission` dans le modèle :

- La méthode `get` : équivalente à la méthode HTML du même nom, elle permet d'envoyer les données depuis l'URL.
- La méthode `put` : propre à XForms, elle permet d'enregistrer les données directement dans la mémoire fichier locale au moyen du système « file : ». Les données sont alors enregistrées dans un fichier XML structuré comme l'instance. Mais cette méthode ne fonctionne qu'en local.
- La méthode `post` : propre à XForms, elle permet d'envoyer les données sous forme de chaîne XML qu'il suffit de récupérer, en PHP, avec la variable globale `$_HTTP_RAW_POST_DATA`.

Il est possible, dans le modèle, d'ajouter des conditions et des contraintes sur certains champs du formulaire XForms. Ainsi, l'élément `bind` dans le modèle permet de lier un champ identifié grâce à l'attribut `nodeset` à une contrainte et à une condition marquée par l'attribut `relevant`.

```
<xforms:bind nodeset="instance('workInstance')/information/level[@nb = 1]"
readonly="false()"
relevant="instance('workInstance')/information/level[@nb = 0]/. != ''"/>
```

• L'interface utilisateur

L'interface correspond à la structure des champs du formulaire : zone de texte, bouton, cases à cocher, liste déroulante... Tous les types de champ HTML possèdent leur équivalent en XForms et je ne vais donc pas les énumérer ici.

Parmi les interfaces « nouvelles », on peut remarquer l'élément `output`, qui permet d'afficher simplement la valeur d'un champ sans possibilité d'écriture. Cet élément est très pratique pour montrer un aperçu du formulaire à l'utilisateur. De plus en ajoutant l'attribut `incremental="true"` à chaque champ, l'aperçu se fera en temps réel.

Enfin, il existe l'élément `upload`, qui permet à l'utilisateur de choisir un fichier via une boîte de dialogue. Il est alors possible d'uploader le fichier choisi grâce à un élément `trigger` qui permet de déclencher des événements. Cet élément nécessite de déclarer le type du champ comme une URI et requiert la méthode de soumission `form-data-post` qui correspond en HTML à la méthode `post` avec `enctype="multipart/form-data"`.

```
<xforms:bind nodeset="instance('documentInstance')/text"
type="xs:anyURI" />
<xforms:submission action="admin.php?action=image"
method="form-data-post" id="portraitForm" />
```

• Les traitements

Il est possible avec XForms d'effectuer des traitements en fonction des valeurs des champs choisis par l'utilisateur. Par exemple on peut effectuer une opération sur plusieurs champs pour calculer automatiquement la valeur d'un autre champ. (N'ayant pas eu besoin d'utiliser de traitements, je n'ai pas approfondi cette partie.)

2.5.4 – Choix de conception

On peut, grâce à l'attribut `src`, charger directement une instance à partir d'un fichier XML. Il est donc possible de charger les informations d'une ressource au format XML ce qui permet de la modifier.

Cela permet d'utiliser le même formulaire pour un ajout et pour une modification de ressource. Dans le premier cas, on charge simplement un fichier XML vide au format de la ressource. Pour une modification, on utilise le fichier XML de la ressource, ce qui permet de pré-remplir tous les champs pour pouvoir ensuite les modifier.

Pour cela, il faut changer dynamiquement la valeur de l'attribut `src` et lui donner le nom du fichier vide ou du fichier de la ressource choisie. J'ai donc décidé de placer les formulaires XForms dans des fichiers XML, ce qui permet donc de détecter le nœud XML correspondant à l'instance XForms et de modifier l'attribut grâce aux fonctions PHP de l'API simpleXML.

Enfin, en ce qui concerne la soumission de données, le mieux aurait été d'utiliser la méthode `put` pour créer directement un fichier XML au bon format. Mais cette méthode ne fonctionnant qu'en local, j'ai préféré utiliser la méthode `post` couplée à une fonction PHP récupérant la chaîne XML et l'insérant dans un fichier XML préalablement créé.

2.5.5 – Difficultés rencontrées avec la technologie XForms

La technologie Xforms étant très récente, les « plugins » XForms pour les navigateurs ne sont pas encore complètement développés et il persiste plusieurs bogues et dysfonctionnements. Le plus gros problème avec XForms a été d'en faire fonctionner un exemple. J'ai choisi d'utiliser un « plugin » pour Mozilla Firefox, celui pour Internet Explorer étant payant.

Comme il existe encore peu d'aide et d'explication sur cette technologie et qu'elles sont la plupart du temps en anglais, j'ai décidé de tester l'un des rares exemple que j'ai pu trouver sur Internet.

J'ai pu me rendre compte que la plupart des exemples étaient incomplets et que XForms demandait beaucoup de rigueur notamment au niveau de l'utilisation des espaces de noms (namespace). Cependant, il est probable qu'à l'avenir XForms ne nécessitera plus de namespace. De la même façon, un formulaire XForms devait obligatoirement se trouver dans un fichier XHTML pour s'afficher correctement.

3. Développement de la partie navigation

3.1 – Principe

Le principe de la navigation est simple : il s'agit d'effectuer une recherche pour obtenir une liste d'œuvres ou de compositeurs, répondant aux critères choisis, puis de cliquer sur l'une des ressources de la liste pour avoir accès aux informations la concernant.

The screenshot shows the Ircam website interface. At the top left is the Ircam logo (Centre Pompidou). The main header area contains the word 'Brahms' and navigation links: 'Accueil | Recherche | Contact'. Below this is a banner for 'Encyclopédie multimédia de l'Ircam sur la musique contemporaine'. The main content area is divided into two columns. The left column, titled 'Navigation', contains search filters for 'Compositeurs' (Nom, Année de naissance, Pays, etc.) and 'Oeuvres' (Titre d'oeuvre, Date de composition, etc.). The right column, titled 'COMPOSITEURS PAR ANNÉE DE NAISSANCE', has a search box with '1945' entered and an 'ok' button. Below the search box, it displays '30 Résultat(s) pour 'Année de naissance' = 1945' and a list of names including Amirkhanian Charles, Aperghis Georges, Arrachart Jean-Marc, etc.

Figure 6 : exemple d'une recherche

La navigation est donc composée d'un moteur de recherche permettant d'accéder à différentes ressources. On peut ensuite accéder aux ressources liées, ce qui permet de naviguer dans la base.

3.2 – Conception des interfaces

3.2.1 – Interface avec la base de données

Comme nous l'avons vu précédemment, la base de données est une architecture SQL couplée à quelques fichiers XML. Pour traiter les informations provenant de cette base de données, le système utilise des fonctions PHP permettant de récupérer les données SQL et de les structurer au format XML.

Pour simplifier les accès entre les fonctions PHP et la base SQL, une classe Mysql a été conçue. Elle constitue une interface de communication avec la base de données. Cette classe facilite la création, l'envoi et le retour des résultats pour l'ensemble du système.

La classe Search communique avec la base de données SQL à l'aide de requêtes. Cette classe définit les modalités de recherche. Elle retourne le résultat d'une recherche en fonction du critère demandé.

Enfin, les classes Person, Work et Document, héritant de la classe Ressource, s'occupent de récupérer les informations de la base de données SQL et des fichiers XML, et écrivent les données dans un fichier XML temporaire au format de publication.

3.2.2 – IHM de navigation

Après traitement de la base de données par les classes PHP, nous disposons d'un fichier XML temporaire. La solution la plus simple pour en afficher les données est l'utilisation de feuilles de style XSL permettant d'exporter le fichier XML en un contenu XHTML. La mise en page du code XHTML est assurée par une feuille de style CSS.

Biographie	LIGETI, GYÖRGY										
Cité dans les documents <ul style="list-style-type: none"> ▣ Biographie de Antoine Bonnet ▣ Biographie de Luis De Pablo ▣ Biographie de Gérard Grisey ▣ Biographie de Michaël Lévinas ▣ Bio développée de Luis De Pablo ▣ Bio développée de Franco Donatoni ▣ Bio développée de Klaus Huber ▣ Bio développée de Tristan Murail ▣ notice de Congrueces 	Information sur le document <table border="1"> <tr> <td>Auteur</td> <td></td> </tr> <tr> <td>Langue</td> <td>Français</td> </tr> <tr> <td>Categorie</td> <td>Biographie</td> </tr> <tr> <td>Type de media</td> <td>texte</td> </tr> <tr> <td>Source</td> <td>Programme du Théâtre du Châtelet 1996-1997</td> </tr> </table>	Auteur		Langue	Français	Categorie	Biographie	Type de media	texte	Source	Programme du Théâtre du Châtelet 1996-1997
	Auteur										
	Langue	Français									
	Categorie	Biographie									
	Type de media	texte									
	Source	Programme du Théâtre du Châtelet 1996-1997									
		Contenu <p>Né le 28 mai 1923 à Dicsőszénmárton (Transylvanie), György Ligeti effectue ses études secondaires à Cluj où il étudie ensuite la composition au Conservatoire auprès de Ferenc Farkas (1941-1943). De 1945 à 1949, il poursuit ses études de composition avec Sándor Veress et Ferenc Farkas à l'Académie Franz Liszt de Budapest où il enseigne lui-même l'harmonie et le contrepoint entre 1950 et 1956. Il fuit alors la Hongrie suite à la révolution de 1956 et se rend d'abord à Vienne, puis à Cologne où il est accueilli notamment par Karlheinz Stockhausen. Là, il travaille au Studio électronique de la Westdeutscher Rundfunk (1957-1959) et rencontre Pierre Boulez, Luciano Berio, Mauricio Kagel... En 1959, il s'installe à Vienne et obtient la nationalité autrichienne en 1967.</p> <p>Dans les années soixante, György Ligeti participe chaque année aux cours d'été de Darmstadt (1959-1972) et enseigne à Stockholm en tant que professeur invité (1961-1971). Lauréat de la bourse du Deutscher Akademischer Austausch Dienst de Berlin en 1969-1970, il est</p>									

Figure 7 : IHM de navigation

Pour associer ce fichier XML généré, la classe Xslt définit le format d'export pour la publication. Cette classe se charge d'associer le fichier XML et la feuille de style XSL appropriée, et de retourner le résultat sous la forme d'un contenu XHTML.

Une fois la page XHTML affichée, l'utilisateur a accès aux informations des ressources et dispose de lien vers les objets sous jacents. Il peut ainsi naviguer librement dans la base.

3.2.3 – Schéma conceptuel

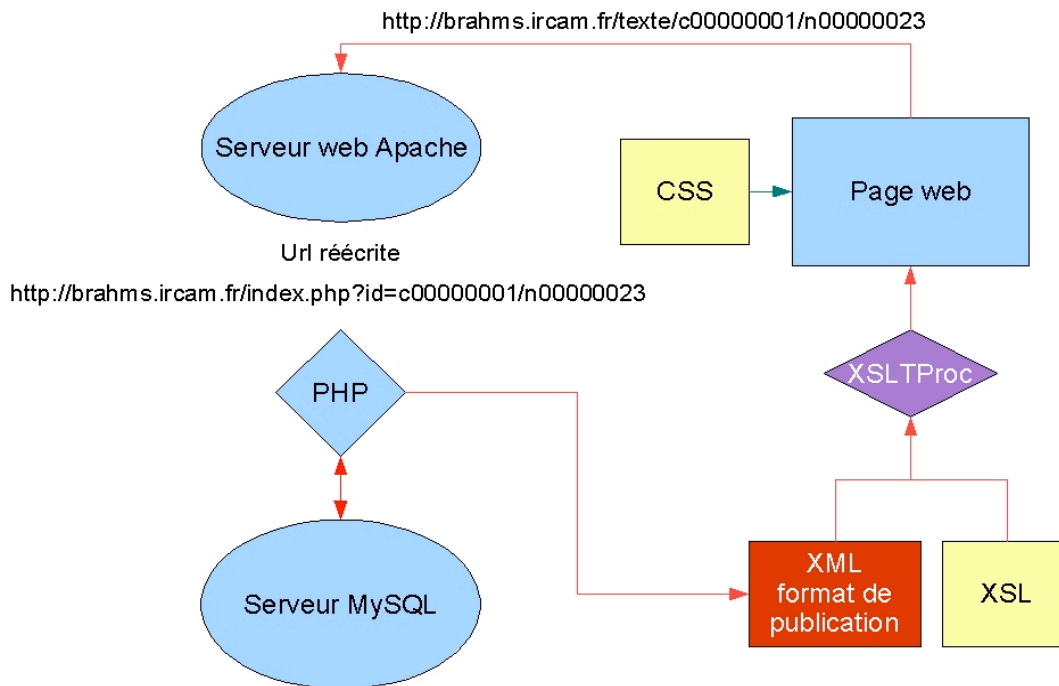


Figure 8 : schéma conceptuel de la partie navigation

Le serveur web reçoit une URL dans un ancien ou nouveau format. Il la réécrit de façon à ce que PHP l'interprète. Une fois la demande identifiée, les classes PHP récupèrent les informations de la base de données puis génèrent un fichier XML temporaire. Celui-ci est ensuite exporté en un contenu XHTML par l'intermédiaire d'une feuille de style XSL puis affiché. Toute cette procédure de publication permet de bien séparer le contenu (SQL/XML) du contenant (XSL/CSS).

3.3 – Spécifications système

Pour fonctionner, la partie navigation a besoin d'un serveur HTTP permettant d'exécuter PHP. Un processeur XSLT, utilisable par les fonctions PHP, est également nécessaire. Il permet de transformer, à l'aide d'une feuille de style XSL, un fichier XML en un code XHTML.

Le serveur utilisé est Apache 2 avec un module PHP version 5 et supportant MySQL 4. Pour la publication des fichiers XML, le processeur XSLT est XSLTProc qui utilise LibXML2 (standard sur les plate-formes Linux).

Pour résoudre les anciens modèles d'URL, utilisés dans la version d'origine, le module de réécriture d'URL d'Apache doit être activé. Il permet de transformer les anciennes URL dans un format actuel.

Les choix se sont portés vers ces modules dans un souci de portabilité : le site doit être accessible sous Windows (Internet Explorer) et Linux (Mozilla, Netscape).

3.4 – Découpage en fichiers

La partie navigation utilise plusieurs fichiers.

D'abord, les fichiers contenant une partie de la base de données :

- Brahms.xsd : contient la DTD,
- un ensemble de fichiers XML contenant le texte des objets Documents,
- un ensemble d'image au format « jpeg » correspondant aux Documents de type Image.

Ensuite, les fichiers d'interface avec la base de données :

- .htaccess : permet la réécriture des anciennes URL,
- Index.php : le « main » des classes PHP (créer les objets PHP en fonction de l'URL),
- mysql.php, search.php, ressource.php, person.php, work.php, document.php : chaque fichier comprenant la classe du même nom.

Quelques fichiers d'interface :

- accueil.xml : contient les « données » de la page d'accueil,
- personSearch.xml et workSearch.xml : contiennent respectivement les formulaires de recherches expertes pour un compositeur et pour une œuvre,
- genre.xml : contient l'arborescence de la liste de tous les genres musicaux.

Les fichiers XSL permettant d'afficher correctement les données :

- menubox.xsl : contient des « templates » pour afficher le menu en fonction de l'objet,
- template.xsl : contient des « templates » pour afficher les éléments communs à toutes les pages (la bannière par exemple),
- ressource.xsl, person.xsl, work.xsl, document.xsl : permet d'afficher les ressources,
- result.xsl : permet d'afficher les résultats d'une recherche,
- standard.xsl : permet d'afficher certains fichiers XML comme la page d'accueil,
- xmlTree.xsl : permet un affichage sous forme d'arbre (utilisé pour les genres).

Enfin, les fichiers permettant la transformation XSLT :

- les bibliothèques XSLTProc, libXML2,
- xslt.php : classe PHP associant les fichiers XML aux feuilles de styles XSL correspondantes.

Puis la feuille de style CSS :

- style.css.

3.5 – Principales difficultés

3.5.1 – Erreurs dans la base de données

La première difficulté a été de corriger les erreurs de la base de données. En effet, la base SQL présentait des incohérences comme des incompatibilités de types, des problèmes d'homogénéisation des champs, et des erreurs de contenu. De plus, les fichiers XML de la base de données étaient, pour certains, malformés :

- aucun contenu textuel,
- problèmes d'encodage,
- erreurs de « balisage »,
- inexistantes dans la base SQL et dans l'ancienne base « Brahms ».

Toutes ces difficultés ont dû être réglées au cas par cas.

3.5.2 – Problème du référencement

Afin de permettre aux moteurs des sites de référencement de référencer effectivement les pages Brahms, il était impératif de fournir des systèmes qui permettent d'accéder à l'ensemble du site par navigation grâce à des liens explicites et non seulement à l'aide d'outils de recherche comme des formulaires. Il a été difficile d'ajouter des critères de recherche permettant d'accéder à l'ensemble des ressources avec un nombre limité de clicks.

Les documents texte stockés en dur posaient aussi un problème de référencement. Effectivement ces fichiers contenaient parfois des liens HTML vers d'autres ressources, d'après l'ancien modèle. Il fallait donc parcourir l'ensemble des fichiers pour remplacer les liens dans le nouveau système. Pour cela j'ai utilisé une fonction PHP pour « parser » tous les fichiers XML et remplacer tous les liens à l'aide de la fonction `preg_replace_callback`.

3.5.3 – Outil de recherche

J'ai rencontré plusieurs difficultés lors du développement du moteur de recherche. Tout d'abord à cause de l'encodage UTF-8. En effet, un caractère UTF-8 a une taille variable, ce qui rend certaines opérations sur les chaînes de caractères plus compliquées : le calcul du nombre de caractères ; le positionnement à une distance donnée dans un fichier texte et en règle générale toute opération nécessitant l'accès au caractère de position N dans une chaîne.

Mis à part les nombreux problèmes de conception de requêtes pour les recherches multicritères, la recherche phonétique a également posé problème. Malgré l'utilisation de fonctions existantes, il s'est avéré que le résultat était trop approximatif et surtout obligeait à tester la prononciation de toutes les langues (la base répertoriant des compositeurs de nombreuses nationalités). Cependant ce test a montré la possibilité du système à intégrer des composants extérieurs. La recherche phonétique sera dans les prochaines priorités de développement de la base.

3.5.4 – Problèmes de présentation

Les différents navigateurs existants n'interprètent pas de la même manière les feuilles de style CSS. Il a été plutôt difficile d'ajuster la CSS afin d'avoir un rendu identique sous Internet Explorer et sous Mozilla. Je ne me suis pas attardé trop longtemps sur ce problème puisque, en tout les cas, la présentation graphique finale sera externalisée dans une agence de graphisme (pour s'intégrer avec la charte graphique de l'Ircam).

4. Développement de la partie administration

4.1 – Principe

La partie administration sert à mettre à jour la base : saisie ou modification de contenus, publication, suppression... Le principe est simple et en deux étapes : la saisie et la validation. Cela permet d'effectuer une mise à jour contrôlée de la base Brahms. La saisie est effectuée par des rédacteurs et les modifications effectuées ne sont publiées qu'après validation par les responsables.

A cette fin, j'ai ajouté sur le site un système d'identification pour les rédacteurs et pour les administrateurs avec une table `user` dans la base de données composée d'un champ `id`, du nom et du mot de passe crypté suivant le codage md5 pour plus de sécurité.

Une fois identifié, un rédacteur aura accès à un nouveau menu et à de nouvelles options lui permettant d'ajouter ou de modifier une ressource, d'ajouter ou de supprimer une relation entre deux ressources, de publier ou de masquer une ressource mais aussi d'accéder à la liste des propositions qui lui auront été retournées.

En effet, toutes ces mises à jour apparaissent sous forme de liste chez un administrateur identifié. Celui ci peut alors regarder les propositions, apporter quelques modifications si nécessaires avant de valider puis de publier la ressource. Mais l'administrateur peut aussi choisir de renvoyer la proposition au rédacteur qui pourra alors la modifier et la renvoyer.

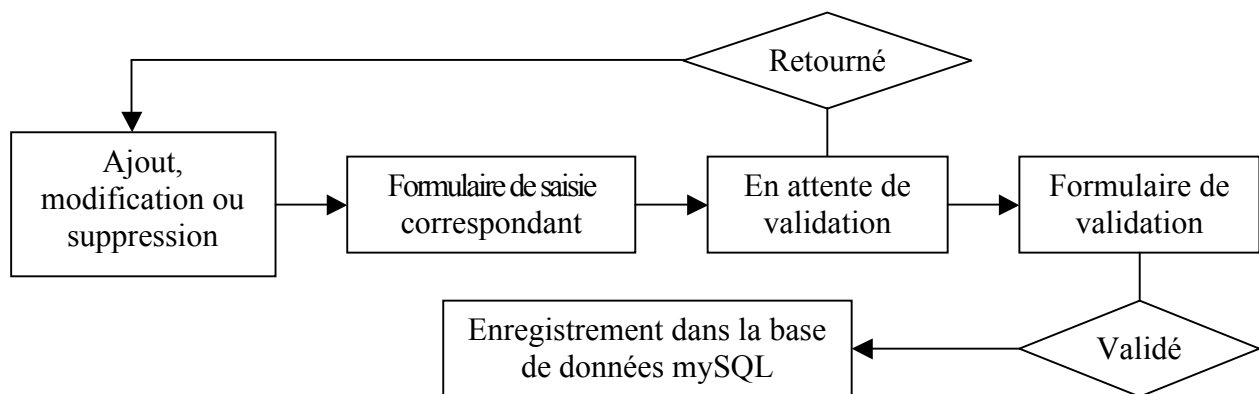


Figure 9 : schéma du principe d'administration

4.2 – Conception des interfaces

4.2.1 – IHM d'identification

L'identification se fait via une « boîte d'identification », apparaissant dans le menu, composée d'un champ texte pour entrer son « login », d'un autre pour saisir son mot de passe et d'un bouton pour valider.

The image shows a web form titled "Identification" in red text. Below the title, there are two input fields. The first is labeled "Login:" in red text, and the second is labeled "Pass:" in red text. Below the "Pass:" field is a button with the text "identifier" in a grey box.

Figure 10 : boîte d'identification

Une fois validé, les informations sont traitées par la classe PHP Session. Elle compare ces informations avec celles contenues dans la table `user` de la base de données pour vérifier si l'utilisateur existe. Si c'est le cas, PHP ouvre une nouvelle session (`session_start()`) et remplit la variable globale `$_SESSION` avec les informations de la personne.

Pour se déconnecter il suffit de cliquer sur un bouton prévu à cet effet. La classe PHP se charge alors de vider la variable de session (`session_unset()`) et de fermer la session (`session_destroy()`).

4.2.2 – IHM d'administration

Il existe deux niveaux d'identification : rédacteur et administration. Un rédacteur effectue des mises à jour qui doivent ensuite être validées par un administrateur. Ainsi, une fois identifié, un rédacteur ou un administrateur accède à un nouveau menu lui permettant de proposer des mises à jour et dispose d'icônes dans la navigation permettant d'agir sur les ressources et les relations.

4.2.3 – Interface entre le système et les formulaires XForms

Comme dit précédemment, une fonction PHP permet de modifier le modèle de données du formulaire XForms de manière dynamique. Il s'agit de la classe Xforms qui manipule les fichiers XML contenant les formulaires et modifie la source de l'instance soit par un modèle « standard » pour un ajout, soit par le fichier XML d'une ressource pour une modification.

Après cette modification, le fichier XML contenant le formulaire XForms est associé à la feuille de style `xforms.xsl` et le formulaire est affiché de la même manière que pour la partie navigation.

Informations générales		
Prénom/Nom	Clément	Caillard
Naissance	1985-03-20	Paris
Mort		
Nationalité	Iles Falkland Iles Féroé Fidji finlandaise française	
Sexe	Homme	
Description	stagiaire IRC&M	

Envoyer Retour

Figure 11 : IHM d'un formulaire de modification de compositeur pré-rempli

Une fois le formulaire rempli et envoyé par un rédacteur, un fichier XML est créé et stocké dans le répertoire « submitted ». Pour générer ce fichier, le système utilise les classes de la navigation auxquels j'ai ajouté la classe Relation qui définit une relation entre deux ressources. L'administrateur peut alors parcourir ces fichiers grâce à la classe FormList, qui permet d'afficher la liste des formulaires envoyés. Cette classe est aussi utilisée pour qu'un rédacteur puisse accéder à la liste des formulaires qui lui ont été retournés (ceux-ci étant stockés dans le répertoire « returned »).

L'administrateur peut alors valider un formulaire qui est alors enregistré dans la base de données grâce aux classes PersonToMySQL, WorkToMySQL, DocumentToMySQL et RelationToMySQL.

4.2.4 – Interface interne aux XForms

Comme dit précédemment, il est possible en XForms d'utiliser plusieurs modèles et plusieurs instances. Les formulaires utilisent ce système pour afficher les listes déroulantes. Ainsi, le contenu des listes déroulantes se trouve dans un fichier XML qui est chargé en tant qu'instance supplémentaire. Ce fichier peut même être créé dynamiquement, ce qui évite de mettre à jour les formulaires.

Grâce à l'élément `copy`, on peut copier un nœud entier d'une instance à une autre, ce qui permet de copier les attributs d'un nœud. On peut ainsi afficher le contenu d'un nœud et copier l'id correspondante.

Pour le formulaire d'ajout ou de modification de Document de type Texte, un bouton permet de parcourir le texte saisi par l'utilisateur et de comparer chacun des mots avec les noms de compositeurs et les titres des œuvres. Ce bouton fait appel à la classe DocumentParse qui, si elle rencontre un nom ou un titre, le transforme en lien HTML vers la ressource en question.

4.2.5 – Schéma conceptuel

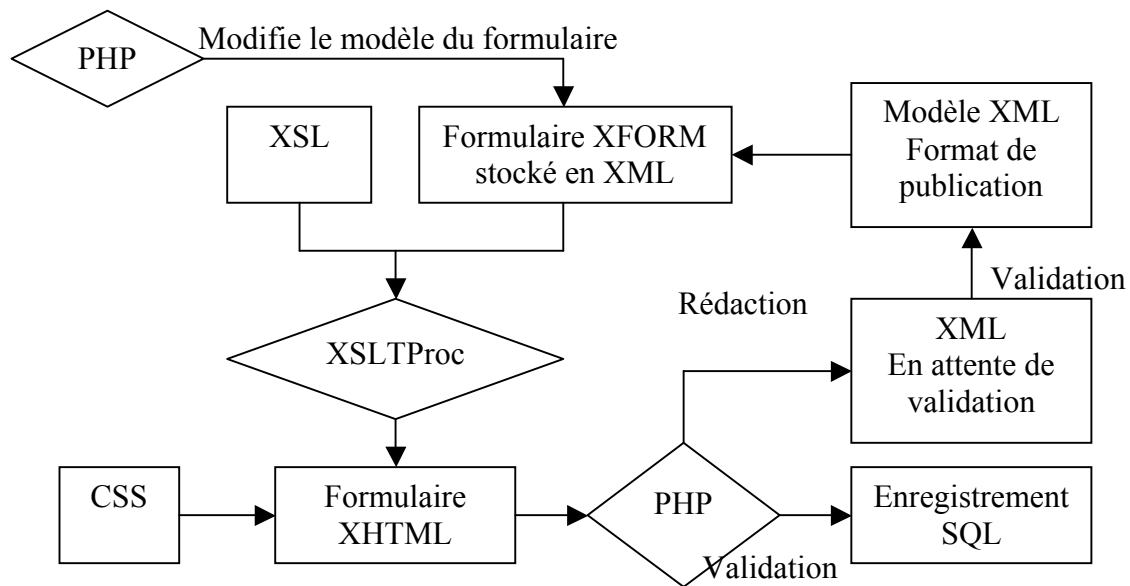


Figure 12 : schéma conceptuel de la procédure d'administration

En résumé, la fonction XForms se charge d'abord d'identifier le type de mise à jour puis de modifier le modèle du formulaire correspondant. Ensuite le formulaire est associé à une feuille de style XSL et transformé en page XHTML mis en forme avec du CSS. Le formulaire est rempli par l'utilisateur et validé. Des fonctions PHP se chargent ensuite, soit d'enregistrer les informations dans la base de données en cas de validation, soit de copier le fichier XML, résultant du formulaire, dans le répertoire adéquat.

4.3 – Spécifications système

L'outil d'administration étant intégré au site et à la navigation, il nécessite les mêmes spécifications que la partie navigation. Autrement, le navigateur utilisé pour accéder à la partie administration doit supporter la technologie XForms.

4.4 – Découpage en fichiers

La partie administration utilise plusieurs fichiers supplémentaires.

D'abord, les fichiers d'interface avec XForms :

- admin.php : le « main » de la partie administration,
- session.php, caddie.php, user.php, relation.php, documentParse.php, formList.php, personToMySQL.php, xforms.php, workToMySQL.php, documentToMySQL.php, relationToMySQL.php : chaque fichier comprenant la classe du même nom
- personForm.xml, documentForm.xml, portraitForm.xml, workForm.xml, personValid.xml, workValid.xml, documentValid.xml, relationForm.xml : les formulaires d'administration XForms.

Puis, les instances de données des formulaires XForms :

- person.xml, work.xml, document.xml et relation.xml : les fichiers d'instance standard,
- composer.xml, instrumentation.xml, nationality.xml, predicat.xml, publisher.xml, et workTitle.xml : les fichiers d'instance externe.

La feuille de style XSL:

- xforms.xsl.

4.5 – Principales difficultés

4.5.1 – Gestion des relations et publication de données

La gestion des relations a posé quelques problèmes de conception pour l'administration. Il y avait plusieurs possibilités et, comme de nouveaux objets risquaient d'apparaître dans la suite du développement, j'ai décidé de faire un formulaire d'ajout de relation globale permettant de rentrer les `id` de deux ressources ainsi que le type de prédicat. J'en ai donc profité pour ajouter la classe Relation (relation.php) qui permettait de générer un fichier XML d'une relation.

Pour faciliter l'utilisation de ce formulaire, j'ai ajouté un système de « caddie » ou de « panier » à la navigation qui permet d'afficher dans un « POPUP » l'`id` d'une ressource et un label explicite (Il s'agit de la classe Caddie contenu dans le fichier caddie.php). Ce système évitait d'utiliser des listes déroulantes contenant les noms de compositeurs ou les titres d'œuvres, ces listes étant bien trop grandes pour être manipulées.

Le système de publication des ressources a posé des difficultés de gestion de la publication pour les ressources sous-jacentes. Après réflexion avec Samuel Goldzmidt, nous avons décidé de masquer ou publier également toutes les relations sous-jacentes d'une ressource.

4.5.2 – Problèmes avec XForms

L'utilisation en général de XForms et le développement des formulaires ont nécessité beaucoup de recherches et de tests. Il y a encore peu de documentation sur cette technologie et la plupart des exemples qu'on peut trouver sur Internet ne sont pas tout à fait exacts. L'utilisation de l'élément `upload`, de plusieurs modèles communiquant ensemble, ou encore la gestion des événements, des contraintes et des conditions, ont été la source de nombreuses difficultés et pertes de temps.

4.5.3 – Problèmes d'affichage des formulaires XForms

L'affichage des formulaires XForms a été anormal. En effet, malgré la transformation XHTML du fichier XML contenant le formulaire, l'interface de saisie n'apparaissait pas. Cependant, le simple fait de copier le code source de cette page dans un fichier XHTML, puis de l'ouvrir permettait d'afficher normalement le formulaire XForms.

Après plusieurs recherches infructueuses, j'ai décidé de développer une fonction PHP pour effectuer cette procédure. Cette fonction récupère donc le contenu de la page et l'écrit dans un fichier XHTML temporaire puis redirige la navigation vers ce fichier. Ceci constitue une solution fonctionnelle mais temporaire en attendant de trouver l'origine du problème.

BILAN

1. Etat du projet

A ce jour, j'ai réalisé une version fonctionnelle du site « Brahms » qui est désormais consultable en interne. Grâce à mon travail, la base « Brahms » peut enfin être mise à jour régulièrement.

Pour tout ce qui concerne le fonctionnement de la partie navigation et de la partie administrative du site, j'ai livré un manuel d'utilisation du site et une analyse technique du projet. Les parties navigation et administration sont entièrement finalisées.

D'un point de vue purement technique, toutes les contraintes du cahier des charges ont été respectées. Le planning a été bien suivi mis à part un léger décalage de quelques jours pour apporter une amélioration au site qui n'avait pas été prévue.

L'Ircam continue de développer le site et d'apporter des améliorations au système. Il est prévu que la base « Brahms » soit opérationnelle et mise en ligne en janvier 2007.

2. Suites et améliorations possibles

La base « Brahms » est désormais fonctionnelle mais certains objectifs du cahier des charges restent à finir. Par exemple, l'outil de statistiques et la gestion des droits n'ont pu être réalisés par manque de temps puisque de nouvelles fonctionnalités non prévues à l'origine étaient plus urgentes aux yeux du responsable.

En effet, chaque réunion de travail avec l'équipe faisait apparaître de nouvelles demandes ou ajouts de fonctionnalités. D'autre part certaines difficultés imprévues, comme par exemple les erreurs dans la base de données, ont nécessité un long travail.

L'Ircam va donc recruter un développeur pour coder un outil de statistiques, qui permettra aux administrateurs de regarder les usages de la base « Brahms » afin d'améliorer le système en fonction de la demande des visiteurs, et pour s'occuper de la gestion des droits sur les images et le son.

Parallèlement, l'Ircam va faire appel à des personnes compétentes en musicologie pour mettre à jour la base de données en utilisant l'outil d'administration que j'ai développé. Il faudra aussi mettre en conformité la charte graphique de la base avec les autres sites de l'Ircam. Ceci devrait être réalisé par une agence extérieure.

Une nomenclature des genres pour les œuvres devrait être rapidement mis en place. Il est prévu d'entrer dans la base une liste de tous les instruments existants, ce qui permettra au rédacteur de saisir le genre d'une œuvre en choisissant le nombre d'instruments de chaque type, le nombre de pupitres, de chanteurs etc... Ceci permettra au visiteur d'effectuer des recherches d'œuvres par effectifs.

Il est aussi prévu que de nouveaux objets soient ajoutés dans la base Brahms comme par exemple l'idée de notions que l'on pourrait lier à certaines œuvres (Par exemple, on pourrait associer à une œuvre la notion de « musique spectrale »).

Il est même possible à l'avenir qu'un système de « Web services » soit mis en place pour que les visiteurs puissent récupérer le contenu du site sans la forme, afin de l'éditer d'une autre manière. De plus, l'idée de publier la base Brahms sur des plates-formes différentes comme des PDA ou des téléphones portables a été émise.

Enfin, le « projet » le plus important prévu pour la base Brahms est de l'associer avec plusieurs autres bases internes comme externes afin d'en faire une véritable encyclopédie culturelle et universelle de partage de la musique contemporaine. Dans un premier temps Brahms devrait être relié à la médiathèque pour les ressources audio ainsi qu'à la base interne « Musica » sur la documentation des œuvres.

3. Apports pour l'entreprise

L'Ircam est une institution culturelle sous la tutelle du ministère de la culture. L'institut a un objectif de partage et de diffusion musicale à l'échelle mondiale que la base « Brahms » permet d'accomplir. La base « Brahms » est actuellement une référence mondiale.

4. Intérêt personnel

Dans l'ensemble ; ce stage représente une expérience positive et très enrichissante tant sur le plan professionnel que personnel.

L'apport principal de ce stage concerne l'expérience acquise dans le domaine de la conception et du développement informatique ainsi que l'enrichissement des connaissances qui en découle :

- Découverte et apprentissage de nouveaux langages de programmation (XML, XForms...),
- Approfondissement des connaissances sur les serveurs Web (Apache, configuration...),
- Amélioration des compétences en bases de données (mySQL, InnoDB, encodage UTF-8, contraintes...) et en développement Web (PHP, CSS...),
- Découverte de nouvelles problématiques de site Internet (encodage, référencement, compatibilité...),
- Etude d'architecture,
- Etude et travail sur des nouvelles technologies en cours de développement,

- Recherches sur Internet :
 - Documentation sur des langages,
 - Solutions de problèmes de développement (bugs, compatibilité, encodage...).
- Passation du travail aux collègues,
- Retour d'expérience et d'exigence de collègues.

Les autres intérêts, non négligeables, sont plutôt d'ordre organisationnel :

- Savoir s'adapter à une organisation spécifique autour d'un projet pluriannuel complexe,
- Travailler sur un projet d'importance et répartir efficacement le travail,
- Mettre en pratique diverses méthodes et connaissances acquises pendant les études tout en les adaptant à un contexte professionnel particulier.

Grâce à l'administrateur système, j'ai pu prendre conscience de la difficulté de la gestion et la sécurisation d'un réseau, d'autant plus que l'Ircam est relié au réseau européen de la recherche : le réseau RENATER.

BIBLIOGRAPHIE

Pour mes recherches, j'ai emprunté plusieurs livres sur les langages utilisés (PHP, CSS, XML, XSL, XHTML...) à la médiathèque de l'Ircam. En complément, j'ai effectué de nombreuses recherches sur Internet.

Recherche sur la compatibilité XHTML et sur l'encodage UTF-8 :

- <http://www.vulgarisation-informatique.com/passer-xhtml.php>
- <http://www.envrac.org/index.php/2006/03/11/58-un-tutoriel-sur-le-character-encoding>

Documentation sur les différents langages utilisés :

- <http://www.commentcamarche.net/xml/xmlxsl.php3>
- <http://www.pompage.net/pompe/cssdezero-3/>
- http://www.werbach.com/barebones/fr_barebone.html
- <http://fr.php.net/manual/fr/ref.simplexml.php>
- <http://www.commentcamarche.net/php/phpreg.php3>
- http://www.lephpfacile.com/manual_mysql/manual.fr_InnoDB.php
- <http://www.aidejavascript.com/article139.html>
- http://php.developpez.com/faq/?page=sessions#sessions_demarrage
- <http://formation.anaska.com/docs/chap20-livre-php-5-avance.pdf>
- <http://www-lium.univ-lemans.fr/~lehuen/master2/xml/cours/cours3.pdf>
- <http://www.info-3000.com/phpmysql/chaine.php>
- <http://formation.anaska.fr/docs/chap20-livre-php-5-avance.pdf>
- http://perso.univ-rennes1.fr/pascal.aubry/doc/php_variables/

Recherches sur la technologie XFORMS :

- http://www.w3schools.com/xforms/xforms_selections.asp
- <http://www.yoyodesign.org/doc/w3c/xforms1/slice8.html#ui-input>
- <http://www.hn2.org/docs/xforms1/sliceG.html>
- <http://www.yoyodesign.org/doc/w3c/xforms1/slice10.html#action-xmlevents>
- <http://www.w3.org/TR/2006/REC-xforms-20060314/>
- <http://www.mozilla.org/projects/xforms/download.html>
- <http://www.yoyodesign.org/doc/w3c/xforms-for-html-authors/>
- <http://gilles.chagnon.free.fr/cours/xml/xforms.html>
- <http://www.orbeon.com/ops/doc/reference-xforms-ng>
- <http://www.w3.org/2006/Talks/05-26-steven-XForms/>

Annexe 1 : historique de l'Ircam

- 1970** Le Président Georges Pompidou demande à Pierre Boulez de créer et de diriger un institut de recherche musicale associé au futur Centre national d'art contemporain.
- 1973** Mise en chantier du bâtiment conçu par les architectes Renzo Piano et Richard Rogers, situé au pied de l'église Saint-Merri, sous le bassin de la place Igor-Stravinsky.
- 1974** Pierre Boulez présente publiquement les objectifs et les enjeux de l'Ircam.
- 1976** Réalisation du processeur de sons numériques 4A, premier développement scientifique conduit à l'Ircam (équipe animée par Giuseppe Di Giugno).
- 1977** Inauguration et ouverture au public du Centre Georges-Pompidou. Organisation par l'Ircam de "Passage du XXe siècle", cycle de soixante-dix manifestations musicales.
Ouverture du bâtiment réservé à l'Ircam.
- 1978** Premier concert dans l'Espace de projection.
Première session de formation pour les compositeurs.
- 1979** Première version du programme Chant (sons synthétiques générés par modélisation de la voix chantée), ultérieurement complétée par le langage de contrôle Formes (équipe animée par Xavier Rodet).
- 1981** Mise au point du système 4X, dernier maillon de la série de processeurs de sons numériques développée depuis 1976.
Création de la version initiale de *Répons*, dans le cadre du festival de Donaueschingen, première oeuvre de Pierre Boulez réalisée à l'Ircam.
- 1983** "Le concept de recherche en musique", cycle de créations et de conférences.
L'Ircam en réseau informatique international (et sur l'Internet dès 1989).
- 1984** Organisation de l'[ICMC](#), colloque international en informatique musicale. Réalisation du logiciel Iana permettant l'analyse psychoacoustique des sons (équipe animée par Gérard Assayag).
- 1985** Premiers logiciels musicaux pour ordinateur personnel (équipe animée par David Wessel).
- 1986** Première tournée aux Etats-Unis de l'Ircam et de l'[Ensemble Intercontemporain](#).
"Systèmes personnels et informatique musicale" : symposium et cycle de créations de Marc-André Dalbavie, Philippe Durville, Philippe Hurel et Magnus Lindberg.
Publication du premier numéro de la revue *InHarmoniques*.

- 1987** Dixième anniversaire du Centre Georges-Pompidou : présentation d'oeuvres nouvelles de Barry Anderson, François Bayle, George Benjamin, Thierry Lancino, Tod Machover, Philippe Manoury, Michael Obst, Kaija Saariaho, Marco Stroppa, Trevor Wishart.
- 1988** Finalisation du langage de programmation graphique Max, logiciel conçu par Miller Puckette, favorisant l'interaction directe entre interprète et ordinateur, utilisé par Philippe Manoury dans *Pluton*.
Première version de Modalys (ex-Mosaïc), logiciel de synthèse par modèles physiques (équipe coordonnée par René Caussé).
- 1989** Début de la première session du Doctorat en musique et musicologie du XX^{ème} siècle, proposé avec l'Ecole normale supérieure, l'Ecole des hautes études en sciences sociales et le Centre national de la recherche scientifique (responsable Hugues Dufourt).
- 1990** Première session du [Cursus de composition et d'informatique musicale](#), formation théorique et pratique d'une année destinée aux jeunes compositeurs. Inauguration de la tour conçue par l'architecte Renzo Piano, représentant la première phase d'extension des bâtiments de l'Ircam.
- 1991** Mise au point de la Station d'informatique musicale, plate-forme de traitement audionumérique pour la recherche, la création et la production musicales (équipe animée par Eric Lindemann).
- 1992** Laurent Bayle succède à la direction de l'Ircam à Pierre Boulez qui devient directeur honoraire.
Refonte de la ligne éditoriale avec la sortie du magazine *Résonance* et de la collection *Les Cahiers de l'Ircam*.
Première opération portes ouvertes.
- 1993** Mise en place par Vincent Puig et Andrew Gerzso du [Forum Ircam](#) visant à renforcer les liens avec la communauté musicale. Participation à une seconde formation doctorale destinée aux jeunes scientifiques et dénommée [DEA Atiam](#) (Acoustique, traitement du signal et informatique appliquée à la musique) (responsable : Jean-Claude Risset).
Première Académie d'été réunissant 120 musiciens de 24 pays.
- 1994** Lancement de la collection monographique Ensemble Intercontemporain, Ircam et Adès avec les disques consacrés à Michael Jarrell, Denis Cohen et Magnus Lindberg. Création d'une voix de castrat pour les besoins du film *Farinelli* (équipe animée par Xavier Rodet et Philippe Depalle).
Finalisation d'AudioSculpt, logiciel graphique permettant de manipuler le son (équipe animée par Gerhard Eckel).
- 1995** L'Ircam et le [CNRS](#) s'associent autour d'une unité mixte codirigée par Hugues Dufourt et Hugues Vinet.
Finalisation du Spatialisateur, logiciel conçu par Espaces nouveaux et l'Ircam (équipe animée par Olivier Warusfel), destiné à l'acoustique des salles, à l'espace sonore urbain et au concert.
Etudes menées sur la qualité des bruits automobiles pour le compte de Renault-PSA (équipe animée par Stephen McAdams).

- 1996** Inauguration des bâtiments Jules-Ferry et Bains-Douches réaménagés par les architectes Daniel Rubin et Patrick Rubin (atelier Canal). Mise en place de la [Médiathèque](#) (équipe de projet animée par Michel Fingerhut). Diffusion de FTS, version logicielle de la Station d'informatique musicale (équipe animée par François Dechelle). Lancement du projet [Studio en ligne](#), soutenu par les ministères de la Culture et de l'Industrie.
- 1998** Première édition du festival [Agora](#), manifestation pluridisciplinaire associant la musique contemporaine à la danse, au théâtre et au cinéma dans plusieurs lieux parisiens.
Première version de l'environnement de composition assistée OpenMusic, développé par l'équipe Représentations musicales (responsable Gérard Assayag).
- 1999** Finalisation de la première version de [jMax](#), environnement programmable pour la réalisation d'applications musicales et multimédia interactives (équipe dirigée par François Déchelle).
François Raffinot rejoint l'Ircam pour y diriger le nouveau département de création chorégraphique destiné à approfondir les liens entre danse et musiques contemporaines. Il crée *Play-Back* sur une musique commandée au compositeur Edmund J. Campion. Création, au sein du pôle acoustique du département Recherche et développement, d'une équipe consacrée au Design sonore, dirigée par Louis Dandrel.
- 2000** A l'occasion de la réouverture du Centre Pompidou après deux ans de travaux, l'Ircam renoue avec la tradition pluridisciplinaire en participant à l'exposition *Le temps vite !* Le centre Acanthes reçoit l'Ircam pour une série d'ateliers et de cours proposés dans 3 villes européennes de la culture : Cracovie, Villeneuve-lez-Avignon et Helsinki. Participation au nouveau projet européen [WEDELMUSIC](#) (Distribution de partitions et interfaces de navigation) et national [RADIO.THEM](#) (Radio Thématique Personnalisable en Ligne sur le Web).
- 2001** Participation/coordination de nouveaux projets européens et nationaux : [CARROUSO](#) (Création et transmission de scènes sonores holophoniques à l'aide du format MPEG4), [CUIDADO](#) (Technologies d'indexation, de recherche et de traitement par le contenu pour des bases de données musicales et d'échantillons), [ECRINS](#) (Environnement de classification et de recherche intelligente de sons), EDISSOHN (Edition et diffusion de scènes sonores 3D normalisées au format MPEG-4), [LISTEN](#) (Interfaces de réalité augmentée pour le design de parcours acoustique dans des musées). Participation de l'Ircam dans un DESS Jeux vidéo et médias interactifs coordonné par l' e C N A M .
Création de l'opéra de Philippe Manoury, *K...* à l'Opéra Bastille.

- 2002** Bernard Stiegler succède à Laurent Bayle à la direction de l'Ircam. Livraison des [Musique Lab](#), six applications réalisées conjointement par l'Ircam et le Ministère de l'Éducation Nationale.
Première édition des [Résonances](#) (rencontres internationales des technologies pour la musique) ouvertes avec la 3^{ème} conférence internationale [ISMIR](#) (International Conference on Music Information Retrieval).
Participation/coordination de nouveaux projets européens et nationaux : [AGNULA](#) (Distribution de logiciels libres audio sur Linux), [DVD À LA CARTE](#) (Compilation d'archives en ligne à la demande), [DOREMI](#) (Représentation de la directivité des instruments musicaux), [MUSICNETWORK](#) (Réseau d'excellence sur les technologies musicales), [PHASE](#) (Interfaces sonores, visuelles et haptiques pour la découverte de la musique).
- 2003** 1700 établissements scolaires téléchargent *Musique Lab*, un logiciel conçu pour l'enseignement musical.
Mise en place de l'équipe Analyse des pratiques musicales.
Jonathan Harvey créé son *Quatrième quatuor à cordes* dans une coproduction avec les Festivals Ars Musica (Bruxelles) et Ultima (Oslo).
- 2004** L'Ircam coordonne la préfiguration de la chaîne HIFI du futur [SemanticHIFI](#). Mauro Lanza, compositeur issu du Coursus de l'Ircam, réalise une création musicale pour une chorégraphie d'Angelin Preljocaj, en coproduction avec l'Opéra national de Paris.
Lancement de nouveaux groupes de recherche musicale (orchestration, rythme, voix).
- 2005** Première édition de la nouvelle revue annuelle de l'Ircam [L'Inoui](#) incluant un DVD-Rom, avec les Éditions Léo Scheer.
- 2006** Frank Madlener succède à Bernard Stiegler à la direction de l'Ircam.

Source : site de l'Ircam (<http://www.ircam.fr/62.html>)

Annexe 2 : manuel d'utilisation

1. Structure du site

La structure du site est assez simple et se compose d'une bannière en haut, un second élément horizontal en deux parties, un menu à gauche et le contenu principal au centre.

Une bannière (Cf : figure 1) en haut possède des onglets pour retourner à l'accueil, effectuer une recherche croisée et contacter l'administrateur. Un clique simple sur la bannière ramène l'utilisateur à la page d'accueil. La bannière reste inchangée durant toute la navigation.

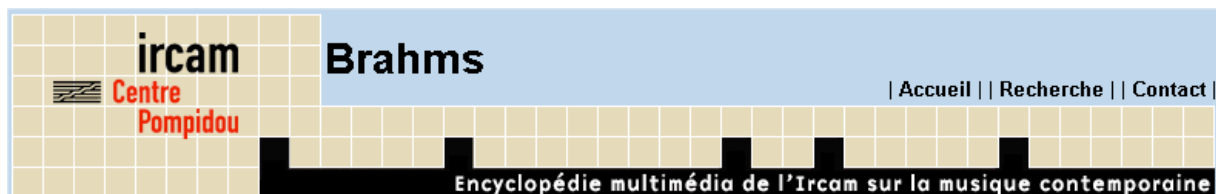


Figure 1: bannière

Le second élément (Cf : figure 2) est composé d'un « carré » sur la gauche que l'on appellera le « type de l'objet » et d'un rectangle à droite appelé « objet » contenant les informations principales du contenu. Le « type de l'objet » indique à l'utilisateur où il se trouve dans la navigation et le type de contenu qu'il est en train de regarder.

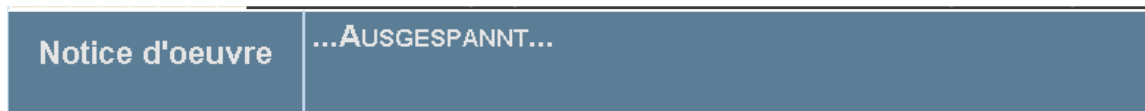


Figure 2: élément objet

Ensuite, on retrouve à gauche un menu variant en fonction de l'endroit où l'utilisateur se situe dans la navigation, mais contenant le plus généralement des liens vers d'autres pages, et à droite le contenu de la page. (Cf : figure 3)

Oeuvres comparables	Informations générales
Genre <input type="checkbox"/> vocale, soliste et ensemble <input type="checkbox"/> voix, soliste et orchestre <input type="checkbox"/> 2 voix d'hommes, clavier solo, et orchestre <input type="checkbox"/> 2 voix d'hommes, orgue solo, et orchestre indéterminé	Compositeur Klaus Huber Titre original ...Ausgespannt... Sous-titre musique sacrée pour baryton, récitant, orgue, cinq groupes instrument et bande magnétique sur des textes de Saint Jean de la Croix, Job, Teilhard de Chardin, Joachim de Fior, Quirinius Kuhlmann
Année <input type="checkbox"/> 1972 (composition)	Année de composition 1972
Editeur <input type="checkbox"/> Schott's Söhne	Genre musical Musique vocale, soliste et ensemble Détail : 2 voix d'hommes, orgue solo, et orchestre indéterminé Instrumentation 2 voix solistes Baryton, récitant Instrument soliste Orgue Ensemble (non précisé) Orchestre Régie son Bande magnétique
	Editeur Schott's Söhne

Figure 3: Menu et contenu

2. Navigation et recherche

2.1 – L'accueil

L'utilisateur entre sur le site par une page d'accueil textuelle (Cf : figure 4) avec un menu de recherche permettant de débiter la navigation dans la base.

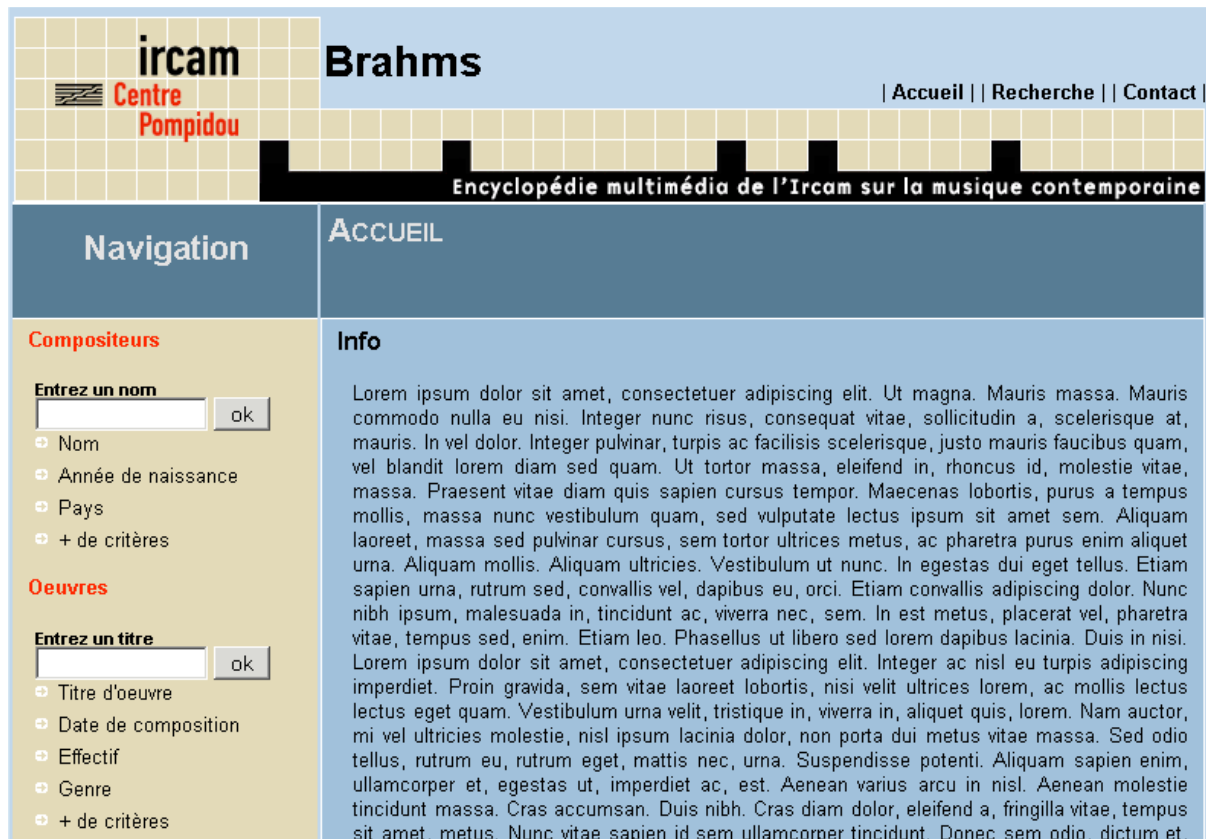


Figure 4: page d'accueil

Le menu de recherche se divise en trois parties : recherche par compositeurs, par œuvres et par œuvres IRCAM, c'est-à-dire les œuvres ayant été créées à l'Ircam.

2.2 – Recherche par compositeurs

Il existe plusieurs critères de recherche pour les compositeurs : par Nom, par Année de naissance et par Pays. On peut aussi effectuer une recherche experte ou croisée en cliquant sur « + de critères ». La recherche experte sera détaillée plus loin. Enfin l'utilisateur a la possibilité d'entrer un nom de compositeur dans le champ texte précédé de la mention « Entrez un nom ».

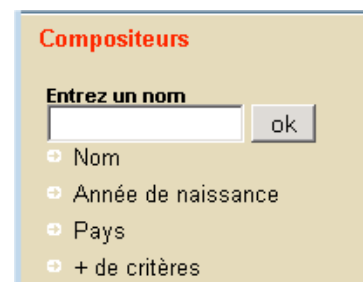


Figure 5: menu de recherche par compositeurs

La recherche par « Nom » est en fait une recherche alphabétique. En cliquant sur ce critère, une page contenant la liste des compositeurs dont le nom commence par un 'A' s'ouvrira et l'utilisateur pourra alors choisir une lettre dans l'élément « objet » du site.

Navigation	NOMS DE COMPOSITEURS PAR ORDRE ALPHABÉTIQUE
Compositeurs Entrez un nom <input type="text"/> <input type="button" value="ok"/> <ul style="list-style-type: none"> <input type="radio"/> Nom <input type="radio"/> Année de naissance <input type="radio"/> Pays <input type="radio"/> + de critères 	Compositeur(s) 208 Résultat(s) pour 'Initiale' = A Aa Michel van der Aaltonen Erkki Aarne Elze Aavik Juhan Abbado Marcello

Figure 6: recherche par Nom de compositeurs

La recherche par « Année de naissance » ouvre une page contenant la liste des compositeurs nés en 1945 et l'utilisateur pourra alors saisir une date dans un champ texte contenu dans l'élément « objet ».

Navigation	COMPOSITEURS PAR ANNÉE DE NAISSANCE
Compositeurs Entrez un nom <input type="text"/> <input type="button" value="ok"/> <ul style="list-style-type: none"> <input type="radio"/> Nom <input type="radio"/> Année de naissance <input type="radio"/> Pays <input type="radio"/> + de critères 	Compositeur(s) 30 Résultat(s) pour 'Année de naissance' = 1945 Amirkhanian Charles Aperghis Georges Arrachart Jean-Marc Arseneault Raynald Barlow Klarenz

Figure 7: recherche par Année de naissance de compositeurs

En cliquant sur « Pays », l'utilisateur sera dirigé vers une page contenant une liste des pays. Ne s'affiche que les pays dans lequel au moins un compositeur, répertorié dans la base, est né. Ensuite, l'utilisateur peut alors choisir un pays pour accéder à la liste des compositeurs nés dans le pays en question.

Navigation	LISTES DES PAYS
Compositeurs Entrez un nom <input type="text"/> <input type="button" value="ok"/> <ul style="list-style-type: none"> <input type="radio"/> Nom <input type="radio"/> Année de naissance <input type="radio"/> Pays <input type="radio"/> + de critères 	Pays 93 Résultat(s) pour 'Pays' = pays Afrique du Sud Albanie Allemagne Argentine Arménie

Figure 8: recherche par Pays

Actuellement, le champ texte invitant l'utilisateur à entrer un nom effectue une recherche selon deux critères. Tout d'abord, le système va chercher tous les compositeurs dont le nom contient la suite de lettre saisie par l'utilisateur, ensuite le système comparera la prononciation phonétique du mot choisi avec celle des noms de compositeurs enregistrés. Néanmoins la recherche phonétique n'est pas au point et est assez approximative.

Ainsi en rentrant « lig », on obtiendra des compositeurs dont le nom contient le segment de lettre « lig » comme « **Holliger** » ou « **Ligeti** » mais aussi des compositeurs dont le début de la prononciation phonétique ressemble à celle de « lig » comme « **Leguay** » par exemple.

Figure 9: recherche libre de compositeurs

2.3 – Recherche par oeuvres

De même, il y a plusieurs critères de recherche pour les oeuvres : par Titre d'œuvre, par Date de composition, par Effectif et par Genre. On peut aussi effectuer une recherche experte ou croisée en cliquant sur « + de critères ». Enfin l'utilisateur a la possibilité d'entrer un titre d'œuvre dans le champ texte.

Notons tout de même qu'il existe aussi une recherche par œuvres IRCAM possédant quelques critères en commun. La seule différence entre les deux recherches est que celle par œuvres IRCAM ne sortira uniquement que les œuvres ayant été réalisées à l'Ircam alors que la recherche normale fournira toutes les œuvres, IRCAM ou non. Autrement, le fonctionnement est similaire et de ce fait, ce manuel ne détaillera que la recherche par œuvres.

Figure 10: menu de recherche par oeuvres

La recherche par « Titre d'œuvre » est aussi une recherche alphabétique fonctionnant de la même façon que la recherche par « Nom » des compositeurs. En cliquant sur ce critère, une page contenant la liste des oeuvres dont le titre commence par un 'A' s'ouvrira et l'utilisateur pourra alors choisir une lettre dans l'élément « objet » du site. L'utilisateur peut aussi cliquer sur « Autres » pour accéder à la liste des œuvres ne commençant pas par une lettre de l'alphabet.



Figure 11: recherche par Titre d'œuvre

La recherche par « Date de composition » ouvre une page contenant la liste des œuvres composées en 1998 et l'utilisateur pourra alors saisir une date dans un champ texte contenu dans l'élément « objet ».

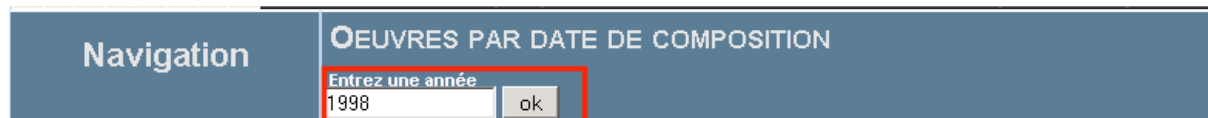


Figure 12: recherche par Date de composition

La recherche par « Effectif » n'est pas encore opérationnelle.

La recherche par « Genre » redirige l'utilisateur vers une arborescence de la liste des genres en 4 niveaux. L'utilisateur peut alors cliquer sur le genre qu'il souhaite pour lancer une recherche.



Figure 13: recherche par Genre

2.4 – Recherche experte de compositeurs

On peut accéder à la recherche experte de compositeurs soit en cliquant sur « + de critères » dans le menu de recherche par compositeurs, soit en cliquant sur l'onglet « recherche » dans la bannière.

De plus, que ce soit dans une recherche experte de compositeurs ou d'œuvres, des onglets vers ces deux types de recherches apparaissent dans l'élément « objet » comme le montre la figure 14 ci dessous.



Figure 14: recherche experte

La recherche experte de compositeurs comprend deux formulaires. Un formulaire de recherche croisée ainsi qu'un formulaire de recherche par anniversaire ou commémoration.

Formulaire

Nom	<input type="text"/>	Prénom	<input type="text"/>
Date de naissance	Jour <input type="text"/> Mois <input type="text"/> An <input type="text"/>	Date de décès	Jour <input type="text"/> Mois <input type="text"/> An <input type="text"/>
Nés entre	<input type="text"/> et <input type="text"/>	Morts entre	<input type="text"/> et <input type="text"/>
Lieu de naissance	<input type="text"/>	Lieu de décès	<input type="text"/>
Pays	<input type="text" value="Indifférent"/> Afghanistan Afrique du Sud Albanie Algérie Allemagne Andorre Angola	Sexe	<input type="text" value="Indifférent"/>
		Modifiés	<input type="text" value="Indifférent"/>
		<input type="checkbox"/> ayant une biographie <input type="checkbox"/> ayant des oeuvres <input type="checkbox"/> ayant une photo	

Figure 15: recherche croisée de compositeurs

En fait, il s'agit d'une recherche sur plusieurs critères. Ainsi, pour la recherche croisée de compositeurs, l'utilisateur peut entrer un nom, un prénom, une date de naissance complète ou non (jour et/ou mois et/ou année), une date de décès sur le même schéma, une fourchette pour les années de naissance et/ou de décès, un lieu de naissance, un lieu de décès, le sexe et le pays d'origine. De plus, l'utilisateur peut effectuer une recherche en fonction de la date de mise à jour en sélectionnant le champs « Modifiés » (Cette année, ce mois, hier ou aujourd'hui). Enfin, l'utilisateur peut aussi demander à ce que seuls les compositeurs possédant une biographie, ayant des œuvres ou ayant une photo ne soient affichés, en cochant les cases correspondantes.

Formulaire2

Année	Jour <input type="text"/> Mois <input type="text"/> An <input type="text"/>	Choix	<input checked="" type="checkbox"/> Naissance <input type="checkbox"/> Mort
Commémoration	<input type="checkbox"/> 5 ans <input type="checkbox"/> 10 ans <input type="checkbox"/> 15 ans <input type="checkbox"/> 20 ans <input type="checkbox"/> 25 ans	<input type="checkbox"/> 30 ans <input type="checkbox"/> 40 ans <input type="checkbox"/> 50 ans <input type="checkbox"/> 100 ans <input type="checkbox"/> tous les 50 ans après 100	

Figure16: recherche par anniversaire et commémoration

Au niveau de la recherche par anniversaire et/ou commémoration, l'utilisateur peut saisir une date complète ou non, et peut cocher l'une ou plusieurs des cases qui sont à sa disposition. Il peut aussi décider de faire une recherche par naissance et/ou par décès en cochant la ou les cases correspondantes.

Ainsi la recherche fournira d'une part la liste des compositeurs nés et/ou morts pendant l'année choisie, d'autre part la liste des compositeurs nés et/ou décédés il y a un certain nombre d'années. Ce nombre étant choisi par l'utilisateur en cochant les cases : 5 ans, 10 ans, ...

Par exemple si l'utilisateur coche les cases « Naissances », « 50 ans » et « 100 ans », le résultat de la recherche affichera la liste des compositeurs nés il y a 50 et 100 ans.

2.5 – Recherche experte d'œuvres

On peut accéder à la recherche experte d'œuvres en cliquant sur « + de critères » dans le menu de recherche par oeuvres.

Formulaire

Compositeur	Nom	<input type="text"/>	Prénom	<input type="text"/>
Titre	<input type="text"/>			
Année de composition	<input type="text"/>	ou entre	<input type="text"/>	et <input type="text"/>
Année de création	<input type="text"/>	ou entre	<input type="text"/>	et <input type="text"/>
Année d'édition	<input type="text"/>	ou entre	<input type="text"/>	et <input type="text"/>
Durée (en minutes)	<input type="text"/>	ou entre	<input type="text"/>	et <input type="text"/>
Editeur	Indifférent <input type="button" value="v"/>			
Information sur la création	<input type="text"/>			
Information sur le commanditaire	<input type="text"/>			
Information sur la dédicace	<input type="text"/>			
Modifiés	Indifférent <input type="button" value="v"/>			
<input type="checkbox"/> ayant une notice <input type="checkbox"/> ayant un livret <input type="checkbox"/> ayant des enregistrements disponibles <input type="checkbox"/> ayant été composées à l'Ircam				
		<input type="button" value="Réinitialiser"/>	<input type="button" value="Chercher"/>	

Figure 17: recherche croisée d'œuvres

Grâce à la recherche experte d'œuvres, l'utilisateur peut effectuer une recherche d'œuvre en fonction de plusieurs critères : nom et prénom du compositeur, titre de l'œuvre, année de composition, de création, d'édition avec possibilité de rentrer une fourchette, durée de l'œuvre, nom de l'éditeur, information sur la création, le commanditaire et la dédicace. De plus, l'utilisateur peut faire une recherche en fonction de la date de dernière mise à jour en sélectionnant le champ « Modifiés ». Enfin, il peut aussi demander à ce que seule les œuvres ayant une note de programme, un livret, des enregistrements disponibles et/ou ayant été composées à l'Ircam soient recherchées.

2.6 – Résultat d'une recherche

Les résultats des recherches apparaissent sous forme de liste de compositeurs par ordre alphabétique, ou de liste d'œuvres par ordre alphabétique, sauf dans le cas d'une recherche par commémoration où le résultat sera divisé en deux parties : la liste des compositeurs en fonction de leur date de naissance et celle en fonction de leur mort. Dans ce dernier cas, des « ancrs » apparaîtront alors dans l'élément « objet ».

Il suffit alors à l'utilisateur de cliquer sur le nom ou le titre de son choix pour accéder à la ressource demandée et à ses informations.

3. Les ressources


Il existe 3 types de ressources différentes : compositeur, œuvre et document. On peut accéder à un compositeur ou à une œuvre grâce à une recherche. Les documents étant liés aux compositeurs ou aux œuvres comme décrit plus loin.

3.1 – Compositeur

Pour un compositeur, le site affichera son nom ainsi qu'un résumé de ses informations dans l'élément « objet » et affichera les informations du compositeur ainsi qu'une photographie éventuelle au niveau du contenu de la page.

KARLHEINZ STOCKHAUSEN
Compositeur allemand né le 22 août 1928 à Mödrath, près de Cologne

Informations générales



Prénom/Nom	Karlheinz Stockhausen
Naissance	22-08-1928, Mödrath, près de Cologne
Nationalité	allemande

Figure 18: compositeur

Ces informations sont suivies par la liste des documents rattachés, la liste des œuvres créées par ce compositeur ainsi que la liste des références de la médiathèque. La page peut alors être assez longue et pour naviguer plus rapidement à l'intérieur, un menu est disponible à gauche contenant des ancres vers chacune des parties.

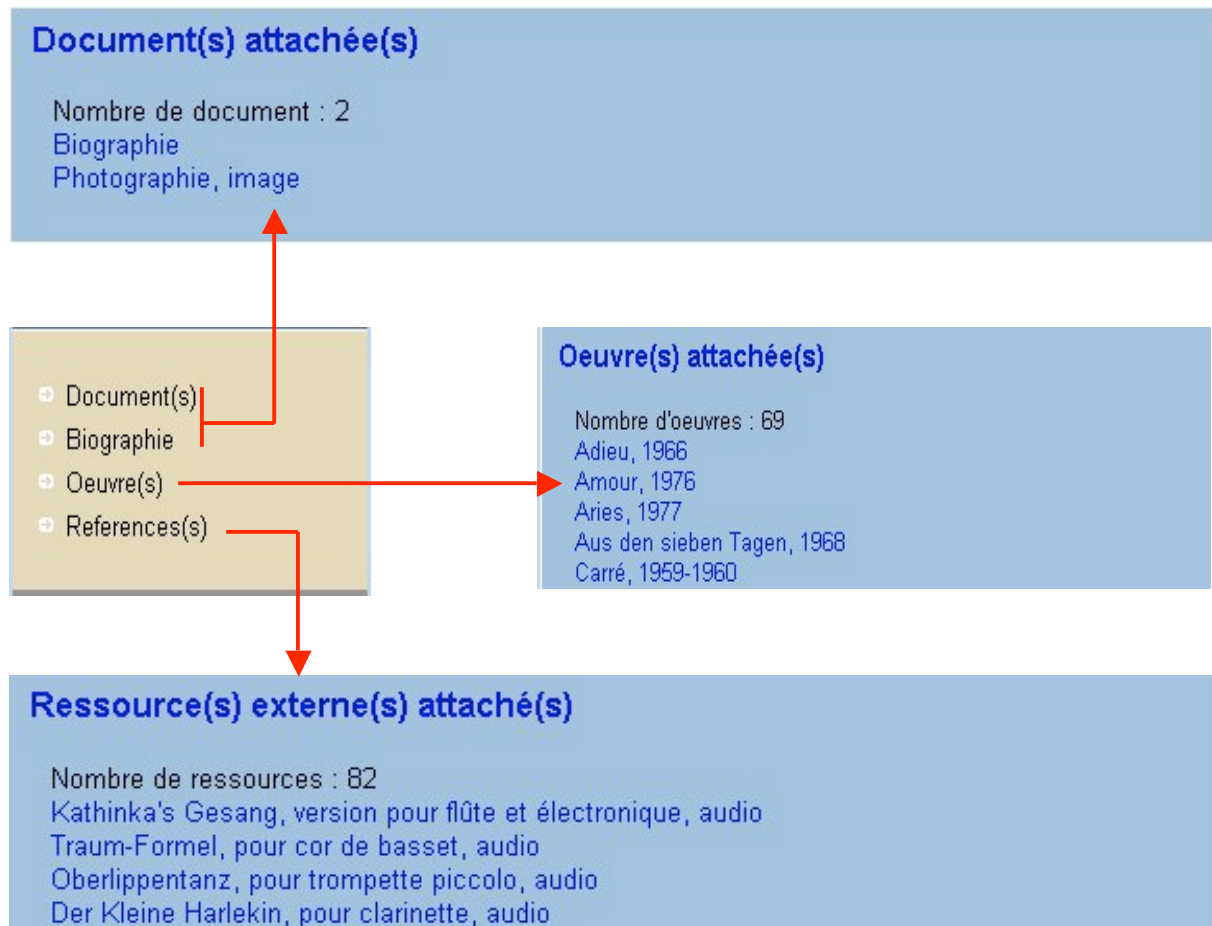


Figure 19: navigation dans une page compositeur

On peut alors cliquer sur une œuvre, un document ou une ressource externe pour regarder les informations de la ressource choisie.

3.2 – Oeuvre

Pour une oeuvre, le site affichera son titre dans l'élément « objet » et affichera les informations au niveau du contenu de la page.

Comme pour un compositeur, ces informations sont suivies par la liste des documents attachés ainsi que le nom du compositeur de l'oeuvre et la liste des références de la médiathèque.

Enfin, un menu permet à l'utilisateur d'effectuer des recherches par genre, similaire à celui de l'oeuvre, par année de composition de l'oeuvre et en fonction de l'éditeur de l'oeuvre.

Notice d'oeuvre	CONCERTO POUR PIANO ET ORCHESTRE												
Oeuvres comparables Genre <ul style="list-style-type: none"> <input type="checkbox"/> concertante <input type="checkbox"/> concerto pour clavier <input type="checkbox"/> concerto pour piano <input type="checkbox"/> concerto pour piano et orchestre Année <ul style="list-style-type: none"> <input type="checkbox"/> 1985 (composition) Editeur <ul style="list-style-type: none"> <input type="checkbox"/> Schott 	Informations générales <table border="1"> <tr> <td>Compositeur</td> <td>György Ligeti</td> </tr> <tr> <td>Titre original</td> <td>Concerto pour piano et orchestre</td> </tr> <tr> <td>Dédicace</td> <td>A Mario di Bonaventura</td> </tr> <tr> <td>Année de composition</td> <td>1985-1988</td> </tr> <tr> <td>Participant(s)</td> <td>Création des 3 premiers mouvements: Anthony di Bonaventura (piano), membres de l'Orchestre philharmonique de Vienne, direction: Mario di Bonaventura Création intégrale: Anthony di Bonaventura (piano), Orchestre symphonique de la radio autrichienne (ORF), direction: Mario di Bonaventura</td> </tr> <tr> <td>Genre musical</td> <td>Musique concertante Détail : concerto pour piano et orchestre</td> </tr> </table>	Compositeur	György Ligeti	Titre original	Concerto pour piano et orchestre	Dédicace	A Mario di Bonaventura	Année de composition	1985-1988	Participant(s)	Création des 3 premiers mouvements: Anthony di Bonaventura (piano), membres de l'Orchestre philharmonique de Vienne, direction: Mario di Bonaventura Création intégrale: Anthony di Bonaventura (piano), Orchestre symphonique de la radio autrichienne (ORF), direction: Mario di Bonaventura	Genre musical	Musique concertante Détail : concerto pour piano et orchestre
Compositeur	György Ligeti												
Titre original	Concerto pour piano et orchestre												
Dédicace	A Mario di Bonaventura												
Année de composition	1985-1988												
Participant(s)	Création des 3 premiers mouvements: Anthony di Bonaventura (piano), membres de l'Orchestre philharmonique de Vienne, direction: Mario di Bonaventura Création intégrale: Anthony di Bonaventura (piano), Orchestre symphonique de la radio autrichienne (ORF), direction: Mario di Bonaventura												
Genre musical	Musique concertante Détail : concerto pour piano et orchestre												
Document(s) attaché(s) Nombre de document : 1 Note de programme													
Personne(s) attachée(s) Nombre de personne : 1 Ligeti, György													
Ressource(s) externe(s) attaché(s) Nombre de ressources : 3 Concerto pour Piano, audio Concerto for Piano and orchestra, audio Concerto for piano and orchestra, audio													

Figure 20: oeuvre

3.3 – Document

Il existe plusieurs types de documents : biographie, biographie développée, article, livret français et étrangers, note de programme et photographie. Quel que soit le type de document, le site affichera le titre de l'œuvre ou le nom du compositeur auquel il est attaché dans l'élément « objet » et affichera les informations au niveau du contenu de la page.

Un menu à gauche donnera une liste de liens vers les documents dans lesquels est cité le document en question.

Biographie	LIGETI, GYÖRGY										
Cité dans les documents <ul style="list-style-type: none"> ▫ Biographie de Antoine Bonnet ▫ Biographie de Luis De Pablo ▫ Biographie de Gérard Grisey ▫ Biographie de Michaël Lévinas ▫ Bio développée de Luis De Pablo ▫ Bio développée de Franco Donatoni ▫ Bio développée de Klaus Huber ▫ Bio développée de Tristan Murail ▫ notice de Congruences 	Information sur le document <table border="1"> <tr> <td>Auteur</td> <td></td> </tr> <tr> <td>Langue</td> <td>Français</td> </tr> <tr> <td>Categorie</td> <td>Biographie</td> </tr> <tr> <td>Type de media</td> <td>texte</td> </tr> <tr> <td>Source</td> <td>Programme du Théâtre du Châtelet 1996-1997</td> </tr> </table>	Auteur		Langue	Français	Categorie	Biographie	Type de media	texte	Source	Programme du Théâtre du Châtelet 1996-1997
	Auteur										
	Langue	Français									
	Categorie	Biographie									
	Type de media	texte									
	Source	Programme du Théâtre du Châtelet 1996-1997									
		Contenu <p>Né le 28 mai 1923 à Dicsöszenmárton (Transylvanie), György Ligeti effectue ses études secondaires à Cluj où il étudie ensuite la composition au Conservatoire auprès de Ferenc Farkas (1941-1943). De 1945 à 1949, il poursuit ses études de composition avec Sándor Veress et Ferenc Farkas à l'Académie Franz Liszt de Budapest où il enseigne lui-même l'harmonie et le contrepoint entre 1950 et 1956. Il fuit alors la Hongrie suite à la révolution de 1956 et se rend d'abord à Vienne, puis à Cologne où il est accueilli notamment par Karlheinz Stockhausen. Là, il travaille au Studio électronique de la Westdeutscher Rundfunk (1957-1959) et rencontre Pierre Boulez, Luciano Berio, Mauricio Kagel... En 1959, il s'installe à Vienne et obtient la nationalité autrichienne en 1967.</p> <p>Dans les années soixante, György Ligeti participe chaque année aux cours d'été de Darmstadt (1959-1972) et enseigne à Stockholm en tant que professeur invité (1961-1971). Lauréat de la bourse du Deutscher Akademischer Austausch Dienst de Berlin en 1969-1970, il est</p>									

Figure 21: document

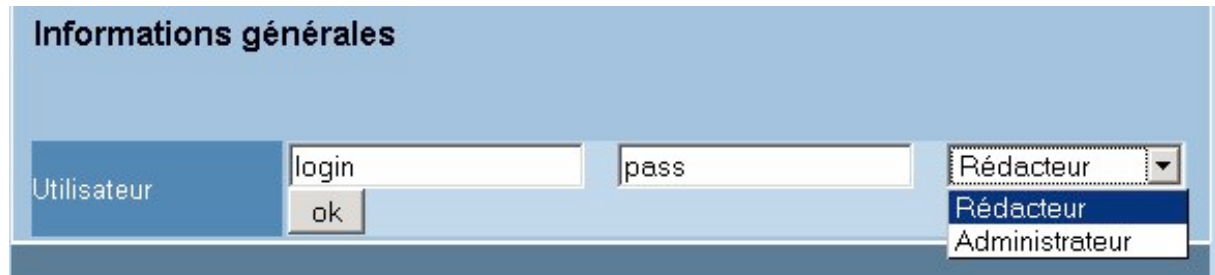
4. Identification / déconnexion

Le site comprend une partie réservée aux rédacteurs et aux administrateurs permettant de mettre à jour la base Brahms. Il est donc possible de s'identifier sur le site. Pour cela, il suffit de rentrer son « login » et son mot de passe dans la petite boîte « Identification » se trouvant dans le menu à gauche. Pour cela, il faut bien entendu avoir été enregistré dans la base de données.


 A screenshot of a web form titled "Identification". It contains two input fields: "Login:" and "Pass:". Below the "Pass:" field is a button labeled "identifier".

Figure 22: boîte d'identification

Seul un administrateur peut enregistrer un rédacteur ou un autre administrateur dans la base. Il doit tout d'abord s'identifier avec ses propres login et mot de passe. Une fois identifié en tant qu'administrateur, la boîte d'identification se transforme alors en un menu pour l'administrateur dans lequel se trouve un lien « ajouter un utilisateur ». En cliquant dessus, il a alors accès à un formulaire d'enregistrement dans lequel il doit rentrer le login et le mot de passe de la personne concernée ainsi que ses droits d'accès (administrateur ou rédacteur).



Informations générales

Utilisateur

login

pass

ok

Rédacteur

Rédacteur

Administrateur

Figure 23: nouvel utilisateur

Une fois identifié, la boîte d'identification se transforme en un menu suivi d'un bouton pour se déconnecter, c'est-à-dire pour retourner sur le site comme si l'on n'était qu'un simple visiteur.



Figure 24: déconnexion

Le nouveau menu permet aux rédacteurs de proposer aux administrateurs de nouveaux compositeurs, œuvres, documents et relations qui devront être validés par l'administrateur.

5. Rédacteur

Comme dit précédemment, une fois identifié, un rédacteur aura accès à un nouveau menu lui permettant de saisir un nouveau compositeur, une nouvelle œuvre, un nouveau document, « d'up loader » une photographie, de créer des relations entre les différentes ressources de la base, ou encore d'accéder aux formulaires qui lui auront été retournés par l'administrateur, qui aura jugé qu'il manquait des informations.



Figure 25: menu rédaction

Outre ce menu, un rédacteur connecté aura accès à quelques « icônes » tout au long de la navigation sur le site qui lui permettront de modifier une ressource, de supprimer une relation existante, ou de publier ou masquer une ressource de la base.

5.1 – Formulaire compositeur

En cliquant sur le lien « Compositeur » dans le menu « Rédaction », le rédacteur ouvre une page contenant un formulaire dans lequel il peut saisir les informations sur le compositeur qu'il souhaite ajouter.



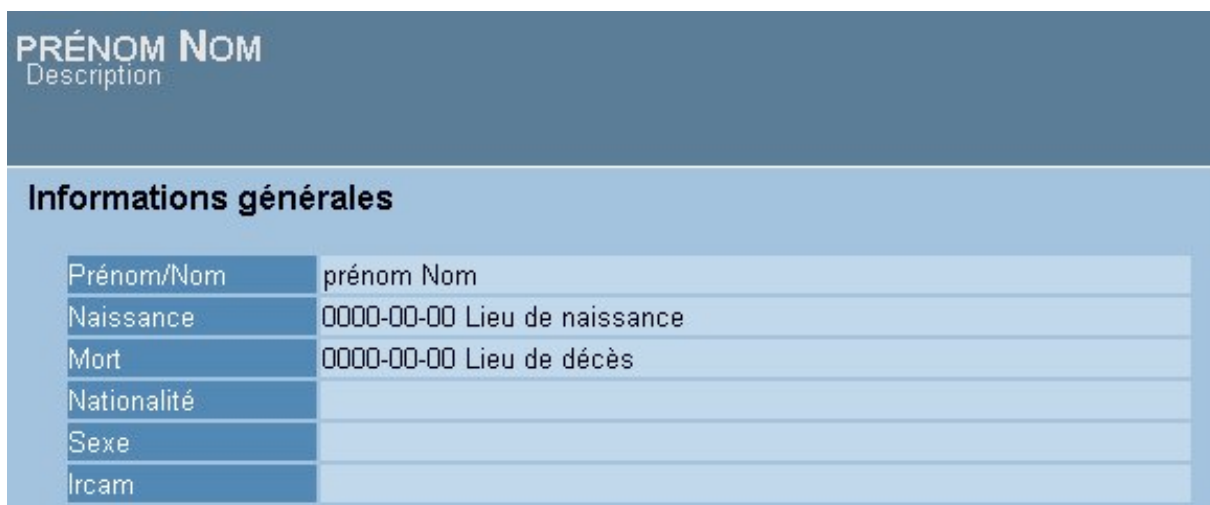
The screenshot shows a web form titled "Informations générales" with the following fields:

Prénom/Nom	prénom	Nom
Naissance	0000-00-00	Lieu de naissance
Mort	0000-00-00	Lieu de décès
Nationalité	antillaise antillaise saoudienne argentine arménienne	
Sexe	<input type="text"/>	
Description	Description	

Buttons: Envoyer, Retour

Figure 26: formulaire de saisi d'un nouveau compositeur

Ainsi le rédacteur pourra remplir les champs : Prénom, Nom, Date de naissance au format AAAA-MM-JJ, Lieu de naissance, Date de décès au format AAAA-MM-JJ, Lieu de décès, Nationalité, Sexe et Description. Il est possible de saisir plusieurs nationalités. La description consiste en un petit résumé des informations. Au fur et à mesure qu'il rentre les informations, un aperçu est mis à jour en temps réel en bas de page.



The screenshot shows a preview of the composer information:

PRÉNOM NOM
Description

Informations générales

Prénom/Nom	prénom Nom
Naissance	0000-00-00 Lieu de naissance
Mort	0000-00-00 Lieu de décès
Nationalité	
Sexe	
Ircam	

Figure 27: aperçu d'un nouveau compositeur

A noter que le Nom est obligatoire. Une fois les informations saisies, le rédacteur doit cliquer sur le bouton « Envoyer » pour soumettre le compositeur aux administrateurs (Cf partie 6.1 sur la validation). Il pourra alors avoir un aperçu tel que le compositeur apparaîtrait s'il était dans la base.

5.2 – Formulaire oeuvre

En cliquant sur le lien « Oeuvre » dans le menu « Rédaction », le rédacteur ouvre une page contenant un formulaire dans lequel il peut saisir les informations sur l'oeuvre qu'il souhaite ajouter.

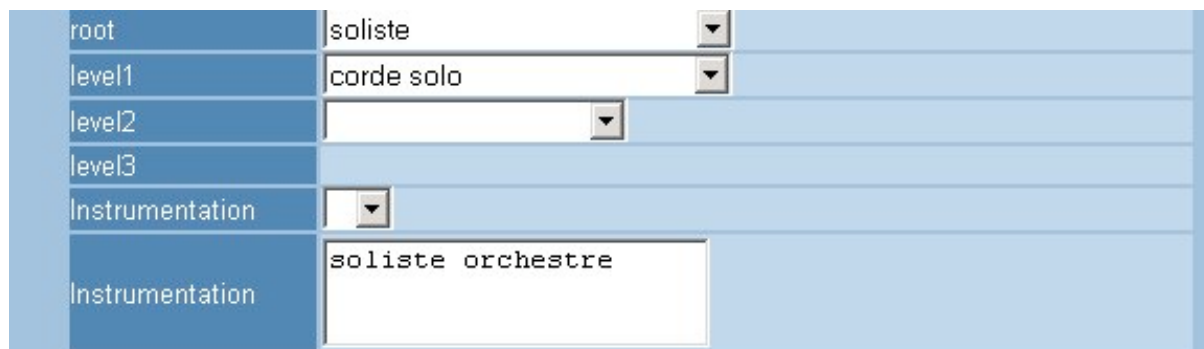
Ce formulaire est assez long et le rédacteur pourra entrer : un titre original, un titre français, un sous-titre, une dédicace, des informations sur la commande, la date de composition, le lieu de création, la date de première création au format AAAA-MM-JJ, des informations sur les participants, des informations sur les lieux, sur les assistants musicaux, une date de fin de composition, la durée, la tonalité, l'Opus, le langage, la section, le code éditeur de l'oeuvre, l'éditeur, le titre de série, le lieu d'édition, la date d'édition au format AAAA-MM-JJ, et la source. De plus, le rédacteur pourra aussi choisir dans le champ « Ircam » si « oui » ou « non » l'oeuvre a été créée à l'Ircam.

Durée	<input type="text" value="00"/>
Tonalité	<input type="text"/>
Opus	<input type="text"/>
Langage	<input type="text"/>
Section(s)	<input type="text"/>
Code éditeur	<input type="text"/>
Editeur	<input type="text"/>
Titre de série	<input type="text"/>
Lieu d'édition	<input type="text"/>
Date d'édition	<input type="text" value="0000-00-00"/>
Ircam	<input type="text" value="Non"/>
Source	<input type="text"/>

Figure 28: formulaire de saisi d'une nouvelle oeuvre

Le rédacteur peut aussi saisir le genre de l'oeuvre. Les genres sont répertoriés dans la base avec 4 niveaux plus l'instrumentation. Le rédacteur devra donc choisir la totalité des niveaux et de l'instrumentation pour déterminer le genre. Il se peut qu'il n'y ai pas besoin de tous les

niveaux pour pouvoir remplir l'instrumentation. En fait, le rédacteur ne peut pas choisir le niveau 2 avant d'avoir rempli le niveau 1 par exemple. Et parfois, le champ instrumentation sera accessible sans que tous les niveaux précédents soient remplis.

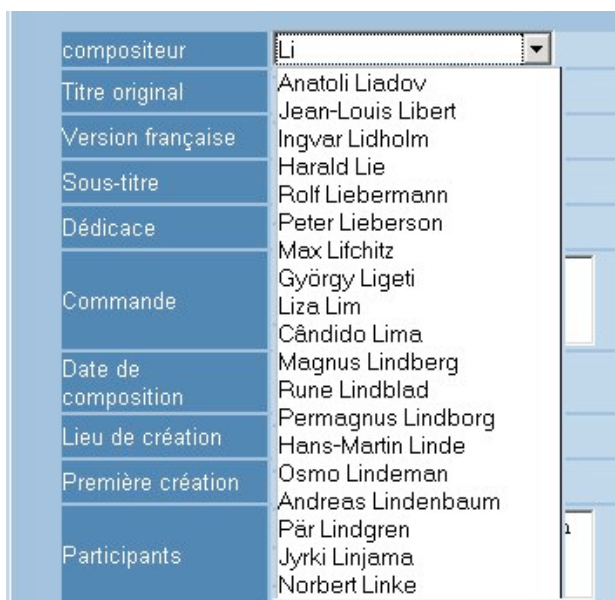


root	soliste
level1	corde solo
level2	
level3	
Instrumentation	
Instrumentation	soliste orchestre

Figure 29: formulaire de saisie d'une nouvelle œuvre (genre)

Par exemple, à la figure 29, on peut remarquer que le rédacteur a choisi les 2 premiers niveaux (root et level1) du genre et qu'il a alors la possibilité de saisir le niveau 3 (level2) du genre mais pas encore le niveau suivant (level3). Par contre il a d'ores et déjà accès au champ « Instrumentation » et peut donc choisir directement une instrumentation. Il pourra ensuite remplir le champ texte pour les détails de l'instrumentation.

Enfin, le rédacteur peut aussi directement choisir le compositeur ayant créé cette œuvre. Pour cela, il doit entrer dans le champ « Compositeur » les deux premières lettres du nom du compositeur pour avoir alors accès à la liste des compositeurs de la base dont le nom commence par ses deux lettres. Il peut alors choisir le compositeur dans la liste.



compositeur	Li
Titre original	Anatoli Liadov
Version française	Jean-Louis Libert
Sous-titre	Ingvar Lidholm
Dédicace	Harald Lie
Commande	Rolf Liebermann
Date de composition	Peter Lieberson
Lieu de création	Max Lifchitz
Première création	György Ligeti
Participants	Liza Lim
	Cândido Lima
	Magnus Lindberg
	Rune Lindblad
	Permagnus Lindborg
	Hans-Martin Linde
	Osmo Lindeman
	Andreas Lindenbaum
	Pär Lindgren
	Jyrki Linjama
	Norbert Linke

Figure 30: formulaire de saisie d'une nouvelle œuvre (relation)

Pour soumettre l'œuvre aux administrateurs (Cf partie 6.1 sur la validation), le rédacteur devra cliquer sur le bouton « Envoyer » après avoir vérifié que le champ « titre original » ait été bien rempli, car obligatoire. Il pourra alors avoir un aperçu tel que le compositeur apparaîtrait s'il était dans la base.

5.3 – Formulaire document

En cliquant sur le lien « Document » dans le menu « Rédaction », le rédacteur ouvre une page contenant un formulaire dans lequel il peut saisir les informations sur le document qu'il souhaite ajouter. Comme pour les autres formulaires, un aperçu est disponible en bas de page.

Informations générales	
Catégorie	Notice
Relation	
Auteur	
Titre	
Source	
Langage	
Contenu	

Figure 31: formulaire de saisi d'un nouveau document

En premier lieu, le rédacteur devra choisir le type de document dont il s'agit : Biographie, Biographie Développée, Notice, Article, Livret français ou étranger.

En fonction de son choix, un champ apparaîtra pour qu'il puisse choisir à quel compositeur ou à quelle œuvre lier ce document. Ce champ fonctionnera de la même manière que le champ « Compositeur » d'un formulaire d'une œuvre. C'est-à-dire que s'il s'agit d'un compositeur, le rédacteur devra d'abord rentrer les deux premières lettres du nom du compositeur pour pouvoir accéder à la liste des compositeurs dont le nom commence par ses deux lettres et s'il s'agit d'une œuvre, le mécanisme est identique sauf qu'il faudra saisir les 3 premières lettres du titre original.

Catégorie	Notice
Relation	App
Auteur	Appalachian spring (7491)
Titre	Apparition de l'Eglise éternelle (10583)
Source	Apparitions (10051)
Langage	Appau (14214)
Contenu	Appello (9721)

Figure 32: formulaire de saisi d'un nouveau document (relation)

Notez que pour les relations avec une œuvre, la liste des œuvres contiendra aussi entre parenthèse l'identificateur de l'œuvre qui permettra au rédacteur de choisir l'œuvre exacte puisqu'il arrive que plusieurs œuvres aient le même titre. Pour trouver l'identifiant d'une œuvre, il suffit de naviguer sur le site jusqu'à l'œuvre recherchée et de prendre le numéro se trouvant tout au bout de l'URL. (ex : http://localhost/brahms/brahms_dev/work/7515 pour Apparition de George Crumb) Une autre manière de récupérer l'identifiant est d'utiliser le système de « caddie » mis à disposition des rédacteurs sur le site (Cf partie 5.7).

Enfin, le rédacteur pourra rentrer les informations du document : l'auteur, le titre, les sources, le langage et le contenu. En ce qui concerne le contenu, le rédacteur est habilité à écrire le texte du document en utilisant des balises HTML tel que `<p>`, `</p>`, `<i>`, `</i>` ou encore ``, ``. Notez que le texte doit toujours être encadré par des balises `<p>` et `</p>`. De plus les balises doivent être correctement indentées.

Ainsi écrire : « Il est né en `<p>` 1945 à Londres `</p>` à l'hôpital de ... » sera mal interprété par le système. Il faudra plutôt écrire : « `<p>` Il est né en `</p>``<p>` 1945 à Londres à l'hôpital de ... `</p>` ». En fait, il faut absolument que tous les mots du contenu soient placés entre des balises `<p>`.

De plus écrire : « `<p>` Il est né `<i>en` 1945`</i>``` à Londres `</p>` » sera mal interprété par le système. Il faudra plutôt écrire : « `<p>` Il est né `<i>en</i>``` `<i>`1945`</i>``` à Londres `</p>` ». Le rédacteur ne doit pas fermer une balise tant que toutes les balises ouvertes plus tôt ne sont pas fermées.

Une fois le contenu rempli, le rédacteur pourra, s'il le désire, cliquer sur le bouton « Liens » pour que automatiquement, tous les noms de compositeurs et titres d'œuvres écrits dans le contenu soient réécrits sous forme de lien vers la base entre balise `<a>`.

Par exemple, si le contenu contient le mot « Apparition », en cliquant sur « Liens », le texte sera transformé en : « ``Apparition`` ».

Pour soumettre le document aux administrateurs (Cf partie 6.1 sur la validation), le rédacteur devra cliquer sur le bouton « Envoyer » après avoir vérifié que les champs « catégorie » et « contenu » aient été bien remplis, car obligatoires. Il pourra alors avoir un aperçu tel que le document apparaîtrait s'il était dans la base.

5.4 – Formulaire image

En cliquant sur le lien « Image » dans le menu « Rédaction », le rédacteur ouvre une page contenant un formulaire dans lequel il peut « up loader » une photographie de compositeur au format jpeg. Comme pour les autres formulaires, un aperçu est disponible en bas de page.

Vous pouvez observer que le formulaire pour les images est quasiment identique au formulaire de document. La seule différence réside dans le contenu. Pour les documents, il s'agit d'un contenu texte alors que pour une image, il s'agit d'un contenu image bien entendu.

Ainsi pour toutes les informations concernant les champs : catégories, relation, auteur, titre, source et langage, se référer à la partie 5.1 sur les formulaires document.

Figure 33: formulaire d'upload d'une image

Pour choisir l'image à up loader, il suffit de cliquer sur le bouton « browse » et de saisir le chemin de l'image sur le disque. Une fois chose faite, et si l'image est bien d'extension « .jpg », un bouton « Upload » apparaîtra. Le rédacteur n'aura plus qu'à cliquer dessus pour envoyer la photographie en validation et en avoir un aperçu.

Figure 34: formulaire d'upload d'une image (upload)

5.5 – Formulaire relation

Le rédacteur peut proposer une nouvelle relation entre deux ressources grâce au formulaire qui lui est proposé lorsqu'il clique sur le lien « Relation ».

Dans ce formulaire, le rédacteur devra choisir le type de relation dans le champ « prédicat » et pourra alors inscrire dans les champs « Sujet » et « Cible » les identifiants des deux ressources. Pour connaître l'identifiant d'une ressource, il est préférable d'utiliser le système de « caddie » mis à disposition lors de la navigation (Cf partie 5.7).

Figure 35: formulaire relation

Il existe sept prédicats différents :

- notice : entre un sujet de type document notice et une cible de type œuvre,
- biographie : entre un sujet de type document biographie développée ou non et une cible de type compositeur,
- article : entre un sujet de type document article et une cible de type œuvre,
- compositeur : entre un sujet compositeur et une cible de type œuvre,
- livret : entre un sujet de type document livret français ou étranger et une cible œuvre,
- portrait : entre un sujet de type document photographie et une cible compositeur,
- refmediathèque : entre un sujet de type référence médiathèque et une cible œuvre.

5.6 – Listes des formulaires retournés

Il peut arriver que les formulaires soumis par les rédacteurs ne soient pas validés par un administrateur et leurs soient retournés pour correction. Dans ce cas là, un rédacteur peut avoir accès à la liste de ses formulaires refusés en cliquant sur « Retourné ». A gauche de chaque ressource est indiqué le type de ressource ainsi que s'il s'agissait d'un ajout ou d'une modification.

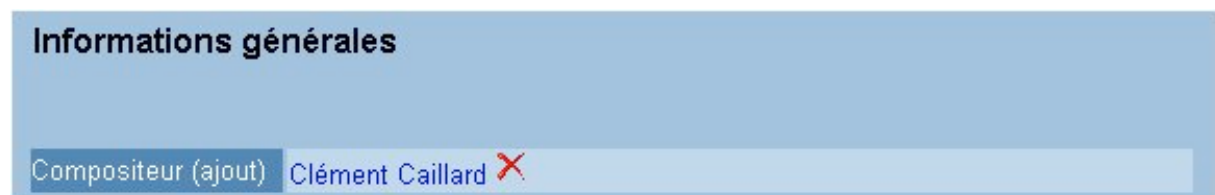


Figure 36: liste des retours

Le rédacteur peut alors cliquer sur un des formulaires retournés pour retomber sur le formulaire avec les informations pré remplies et peut alors apporter des modifications avant de renvoyer le formulaire en validation. Sinon, le rédacteur peut aussi décider de supprimer un formulaire retourné en cliquant simplement sur la petite croix rouge à droite.

Informations générales	
Prénom/Nom	Clément Caillard
Naissance	1985-03-20 Paris
Mort	
Nationalité	française
Sexe	Homme
Description	stagiaire IRC&M

Figure 37: formulaire pré rempli

5.7 – Modification, caddie et publication

Un utilisateur identifié en tant que rédacteur aura aussi accès à plusieurs icônes lui permettant de modifier une ressource, de les « mettre » dans un « caddie » ou « panier », de publier ou masquer une ressource et enfin de supprimer une relation.

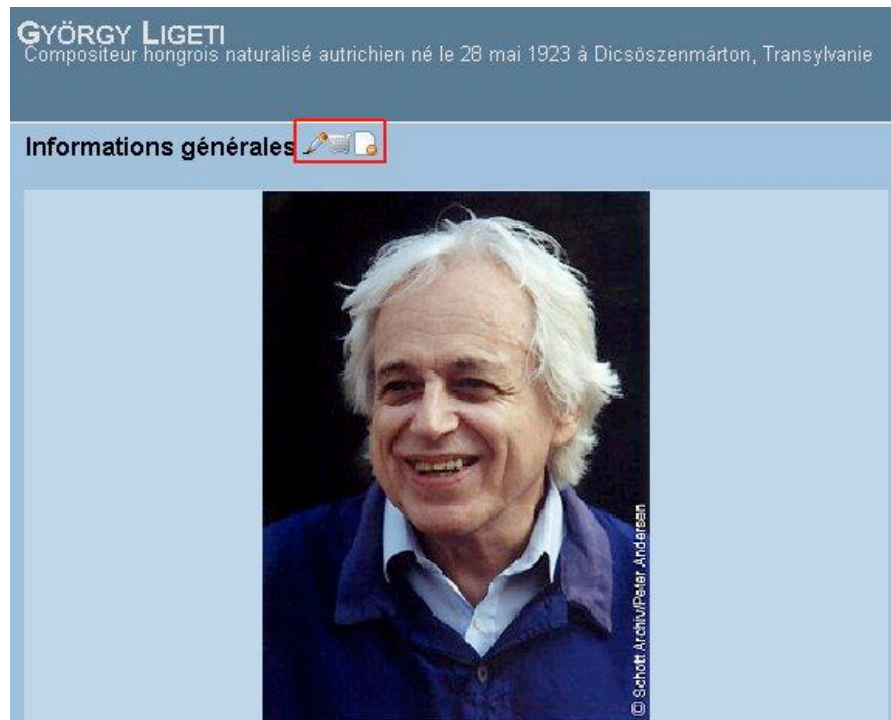


Figure 38: icônes d'un rédacteur

Ainsi en cliquant sur le petit icône en forme de crayon, le rédacteur pourra modifier la ressource en question. Dans ce cas, une page formulaire pré remplie s'ouvrira comme pour les retours. Le formulaire sera tout de même soumis à l'administrateur.

En cliquant sur l'icône en forme de caddie, une nouvelle fenêtre s'ouvrira dans laquelle sera inscrit l'identifiant de la ressource ainsi que ses informations principales comme le nom d'un compositeur ou le titre d'une œuvre par exemple. Plusieurs ressources peuvent ainsi être placées dans le caddie. On peut supprimer une des entrées du caddie en cliquant sur la petite croix rouge à droite des ressources.

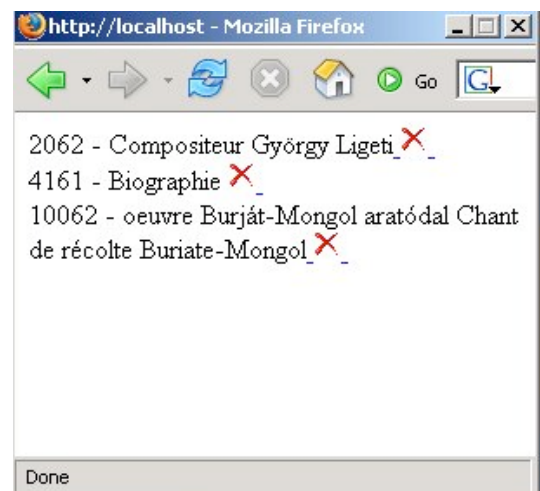


Figure 39: caddie

Le rédacteur a aussi accès à une troisième icône de la forme d'une page avec soit un sens interdit soit une planète qui permettent respectivement soit de masquer la ressource soit de la publier. Une ressource masquée n'est pas visible par les visiteurs simples. Lorsqu'un rédacteur masque une ressource, toutes les ressources attachées comme les documents ou les œuvres sont masqués automatiquement et vice-versa lorsqu'on publie une ressource.

Enfin, le rédacteur voit aussi apparaître un icône en forme de croix rouge à gauche de chacune des ressources liées. Il peut ainsi en cliquant dessus, proposer la suppression du lien entre les deux ressources. Proposition qui sera alors soumise à un administrateur.

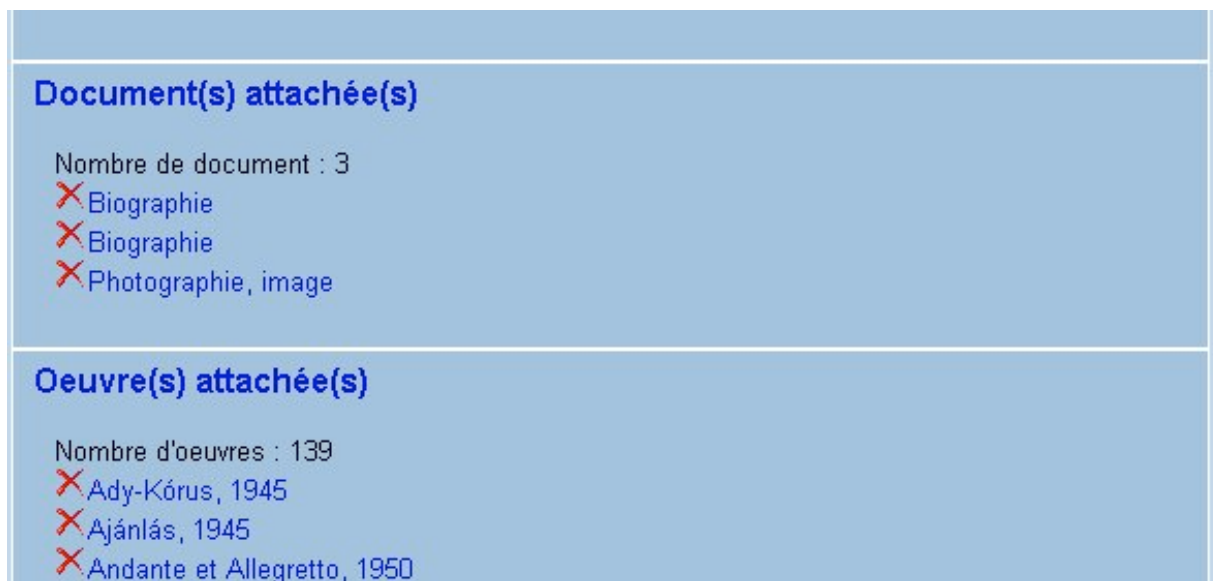


Figure 40: croix de suppression d'une relation

6. Administrateur

Une fois identifié, un administrateur aura accès à un nouveau menu à gauche. Outre les options du rédacteur, un administrateur aura aussi un lien « Validation » et « Ajouter un utilisateur ».

Le lien « Ajouter un utilisateur » lui permettra de créer un nouveau compte rédacteur ou administrateur comme expliqué dans la partie 4.

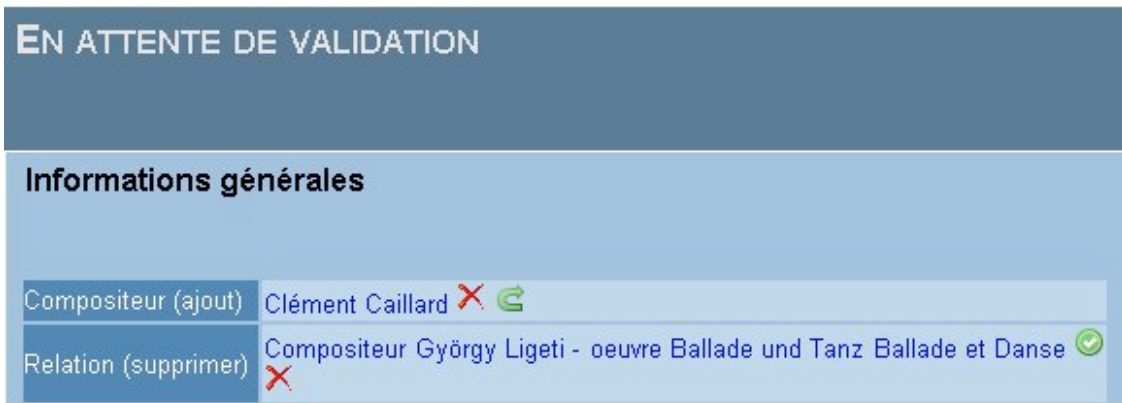
De la même façon, un administrateur aura aussi tous les « icônes » du rédacteur plus une nouvelle icône qui lui permettra de supprimer une ressource.



Figure 41: menu administrateur

6.1 – Validation

En cliquant sur le lien « Validation » dans le menu « Administration », l'administrateur ouvre une page contenant la liste de tous les formulaires soumis par l'ensemble des rédacteurs.






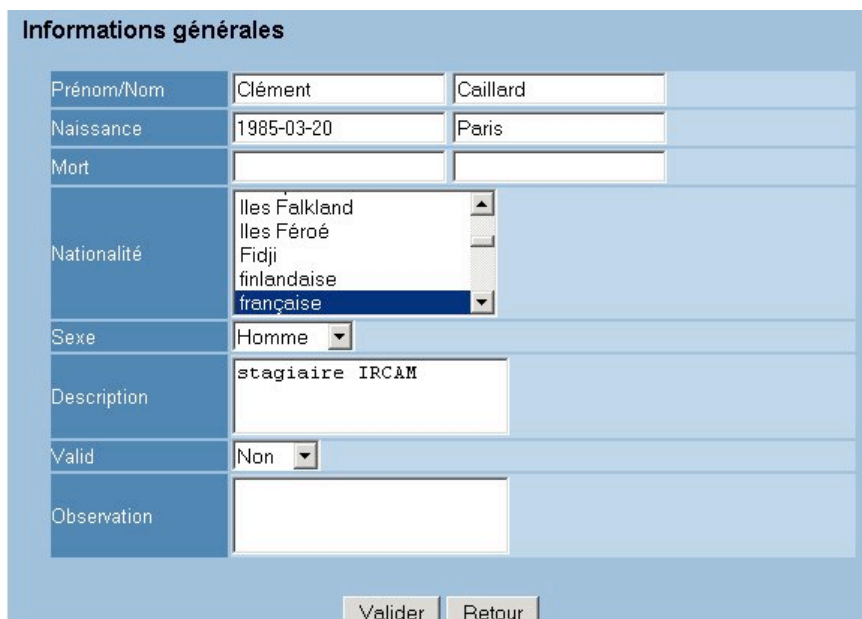
EN ATTENTE DE VALIDATION	
Informations générales	
Compositeur (ajout)	Clément Caillard  
Relation (supprimer)	Compositeur György Ligeti - oeuvre Ballade und Tanz Ballade et Danse 

Figure 42: liste de validation

Cette liste est affichée de la même manière que la liste des retournés (Cf : partie 5.6) à la différence qu'un nouvel icône est apparu en forme de flèche verte faisant « demi-tour » qui permet à l'administrateur de retourner le formulaire au rédacteur. La petite croix rouge permet de supprimer le formulaire.

Pour valider un formulaire de type « compositeur », « œuvre » ou « document », l'administrateur doit simplement cliquer sur le formulaire en question. Cette action aura comme conséquence l'ouverture d'un formulaire de validation pré rempli. Ces formulaires sont pratiquement identiques à ceux d'un rédacteur à la différence que de nouveaux champs sont à remplir comme « valid » et « observation ».



Informations générales	
Prénom/Nom	Clément Caillard
Naissance	1985-03-20 Paris
Mort	
Nationalité	Iles Falkland Iles Féroé Fidji finlandaise française
Sexe	Homme
Description	stagiaire IRCAM
Valid	Non
Observation	

Figure 43: formulaire de validation d'un compositeur

Ainsi le rédacteur pourra vérifier les informations et les modifier si besoin. Ensuite, il pourra choisir si « oui » ou « non » la ressource sera valide une fois enregistrée, c'est-à-dire si elle sera publiée ou masquée. Enfin, l'administrateur pourra ajouter des observations avant de cliquer sur le bouton « Valider » pour que le formulaire soit enregistré dans la base. L'administrateur sera alors redirigé sur la ressource.

En cliquant sur « Retour », l'administrateur reviendra à la liste de validation. Il ne s'agit pas d'un bouton pour retourner le formulaire au rédacteur.

En ce qui concerne les relations en attente de validation, que ce soit pour un ajout ou une suppression, l'administrateur pourra cliquer sur l'icône en forme de rond vert coché pour valider l'ajout ou la suppression. En cliquant sur la croix rouge, il supprimera la proposition définitivement. On ne peut retourner une proposition d'ajout ou de suppression de relation.

Notez que les relations s'écrivent sous la forme « catégorie Ressource1 – catégorie Ressource2 » et qu'il s'agit de deux liens vers les ressources associées. Ainsi l'administrateur pourra aller vérifier simplement de quelles ressources il s'agit.

6.1 – Supprimer une ressource

Outre les icônes du rédacteur, un nouvel icône est disponible pour l'administrateur dans la navigation. Il s'agit d'une croix rouge permettant de supprimer définitivement une ressource de la base. Il faut bien sûr faire très attention avant de l'utiliser et ce n'est pas vraiment conseillé.



BALLADE UND TANZ
Ballade et Danse

Informations générales   

Compositeur	György Ligeti
Titre original	Ballade und Tanz
Version française	Ballade et Danse
Sous-titre	d'après des chants populaires roumains, pour orchestre de jeunes
Année de composition	1949-1950
Genre musical	Musique d'ensemble Détail : octuor bois, percussion, clavier, corde
Instrumentation	octuor divers Flûte à bec soprano, percussions, piano, 3 violons, violoncelle, contrebasse (ad libitum)
Editeur	Schott

Figure 44: suppression d'une ressource