



HAL
open science

DESS ISI Modélisation et implémentation d'une Mémoire Environnementale pour le site de la Mer d'Iroise

Cyril Faucher

► **To cite this version:**

Cyril Faucher. DESS ISI Modélisation et implémentation d'une Mémoire Environnementale pour le site de la Mer d'Iroise. domain_shs.info.gest. 2003. mem_00000187

HAL Id: mem_00000187

https://memic.ccsd.cnrs.fr/mem_00000187v1

Submitted on 17 Dec 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

RAPPORT TECHNIQUE



MISE EN PLACE D'UNE MEMOIRE ENVIRONNEMENTALE DE L'OBSERVATION COTIERE EN MER D'IROISE



Etudiant stagiaire
Cyril FAUCHER

Maitre de stage
Françoise GOURMELON

Tuteur-enseignant
Jean-Yves LAFAYE

LABORATOIRE GEOMER
IUEM 29280 PLOUZANE

DESS ISI - Ingénierie des
Systèmes Informatiques -
UNIVERSITE DE LA
ROCHELLE

22 avril - 19 septembre
2003

DESS Ingénierie des Systèmes Informatiques
Université de La Rochelle
2002/2003

Entreprise d'accueil :
Laboratoire Géomer, **IUEM** (Institut Universitaire Européen de la Mer)
Technopole Brest Iroise – 29280 PLOUZANE

Du **22 avril** au **19 septembre 2003**

Par :
Cyril FAUCHER

Maître de stage :
Françoise GOURMELON

Enseignant tuteur :
Jean-Yves LAFAYE

Septembre 2003.

REMERCIEMENTS

Je remercie tout d'abord Françoise Gourmelon pour son accueil et pour m'avoir permis de travailler sur un projet tel que la "**Mémoire Environnementale**" au sein du laboratoire **Géomer** de Brest.

Je tiens à remercier Jean-Yves Lafaye pour sa contribution dans l'avancement du projet et la validation d'une partie du travail effectué, plus particulièrement au niveau de la spécification du système.

Je remercie Mathias Rouan pour sa disponibilité, son écoute dans le travail et la sympathie dont il a fait preuve pour m'intégrer rapidement dans la vie du labo.

Je salue également les membres du laboratoire **Géomer** : Anne-Sophie, Aurélie, Clotilde, Cyril, Dorothee, Emmanuel, Ingrid, Iwan, Jacqueline, Matt, Solenn, ..., et les personnes interviewées au cours de mon stage pour leur aide et leur soutien, qui ont permis au projet de prendre forme.

Ce projet n'est pas seulement un projet de laboratoire, mais c'est aussi celui de François Cuq, qui souhaitait par celui-ci rapprocher les hommes d'horizons différents et constituer un patrimoine environnemental de la Mer d'Iroise, malheureusement il nous a quitté trop tôt pour en voir l'aboutissement.

SOMMAIRE

SOMMAIRE	3
1. INTRODUCTION	5
2. PRESENTATION DE L'ENTREPRISE	6
2.1. <i>L'entreprise IUEM : Institut Universitaire Européen de la Mer</i>	6
2.1.1. <i>Localisation</i>	6
2.1.2. <i>Histoire</i>	6
2.1.3. <i>Les Services Communs de l'IUEM</i>	7
2.1.4. <i>Organigramme</i>	7
2.1.5. <i>Quelques chiffres :</i>	7
2.1.6. <i>Description de la cellule d'accueil : Géomer UMR 6554</i>	8
2.2. <i>Les moyens informatiques</i>	9
2.2.1. <i>Matériels</i>	9
2.2.2. <i>Outils</i>	9
2.2.3. <i>Le droit de l'informatique au laboratoire Géomer</i>	9
2.3. <i>Le sujet de stage</i>	10
3. PRESENTATION DU TRAVAIL	11
3.1. <i>Analyse du champ de l'étude</i>	11
3.1.1. <i>Description des tâches</i>	11
3.2. <i>Analyse du domaine</i>	14
3.2.1. <i>Définition et limites du domaine</i>	14
3.2.2. <i>La décomposition en packages d'étude pour "mod domaine"</i>	14
3.2.3. <i>Définition des acteurs</i>	15
3.2.4. <i>Construction et description des diagrammes de classes</i>	15
3.2.5. <i>Extrait du dictionnaire de données</i>	30
3.3. <i>Analyse des besoins</i>	35
3.3.1. <i>Schémas des cas d'utilisations (CU)</i>	35
3.3.2. <i>Description générale des C.U.</i>	38
3.3.3. <i>Etude détaillée du C.U. « Gérer un projet »</i>	41
3.3.4. <i>Etude détaillée du C.U. « Consulter une période de projet »</i>	59
3.4. <i>Architecture du système et les interactions</i>	66
3.4.1. <i>Diagramme de déploiement de la maquette</i>	66
3.4.2. <i>Interactions entre les acteurs du domaine et le système</i>	69
3.5. <i>Conception de la base de données</i>	70
3.5.1. <i>L'intégration de la base archi au sein de l'application</i>	70
3.5.2. <i>Schéma de la base de données relationnelle</i>	70
3.5.3. <i>Description des tables de la base de données "archi"</i>	72
3.5.4. <i>Scénario de tests de la base de données</i>	79
3.5.5. <i>Tests d'intégrité de la base de données</i>	81
3.6. <i>Implémentation de deux cas d'utilisations</i>	83
3.6.1. <i>Implémentation du C.U. « gérer un projet »</i>	83
3.6.2. <i>Scénario et tests</i>	84
3.6.3. <i>Implémentation du C.U. « consulter un projet »</i>	86
3.6.4. <i>Scénario et tests</i>	86
3.7. <i>Bilan du projet</i>	87
4. CONCLUSION	89



TABLE DES ILLUSTRATIONS	90
BIBLIOGRAPHIE	91
GLOSSAIRE	92

1. INTRODUCTION



Ce projet intitulé "**Mémoire Environnementale IROISE**" (**MEnIr**), est issu d'une réflexion menée au sein du Comité Scientifique MOTIVE (MOdélisation, Transfert d'Informations, Valorisation pour l'Environnement) du PEVS (Programme Environnement, Vie et Sociétés) du CNRS (Centre National de la Recherche Scientifique) en 2001 et 2002.

Son développement s'inscrit dans le cadre d'un appel à proposition pluridisciplinaire des départements SHS (Sciences de l'Homme et de la Société) et STIC (Sciences des Techniques de l'Information et de la Communication) du CNRS intitulé « Archivage et patrimoine documentaire, apport des sciences de l'information et de la cognition ».

Il apparaît que les phénomènes biologiques, physiques, etc. qui font évoluer notre environnement dépendent d'innombrables paramètres. Afin d'étudier ces phénomènes, il semble donc essentiel de disposer de l'ensemble de ces données pour un site géographique à un instant donné. C'est dans l'idée de rassembler des données provenant de secteurs différents que **MEnIr** trouve sa place en tant que base d'informations pluridisciplinaires.

La réalisation de **MEnIr** demande une réflexion conceptuelle initiale afin d'aboutir à une modélisation du système d'information dans son ensemble, tant du point de vue de ses données que de celui des interprétations qui en sont faites et des questions scientifiques associées, le tout dans une perspective temporelle longue. Des systèmes d'archivage, de gestion et d'analyse doivent être mis en place et testés pour permettre d'accéder aux différentes bases de données et systèmes d'informations en prenant en compte leur évolution dans le temps.

Dans le cadre de la pluridisciplinarité du projet, un partenariat avec le laboratoire d'informatique (L3i) de La Rochelle a été établi. Ce partenariat a déjà démontré son intérêt et son efficacité lors d'un précédent projet (DAHu).

Ce document présente une partie de la conceptualisation du système d'informations **MEnIr**, la construction d'une base de données relationnelle qui en découle et l'implémentation de quelques cas d'utilisations afin de constituer une première maquette de la "**Mémoire Environnementale**". Avant d'entrer dans le vif du sujet, nous ferons connaissance avec l'équipe au sein de laquelle ce projet se déroule.

2. PRESENTATION DE L'ENTREPRISE

Cette partie présente l'environnement de travail dans lequel j'ai évolué durant les cinq mois de stage ainsi qu'une description du sujet de mon stage.

2.1. L'entreprise IUEM : Institut Universitaire Européen de la Mer

2.1.1. Localisation

Technopole BREST-IROISE place Nicolas Copernic 29280 PLOUZANE.



Figure 1 - Trait de côte de la Mer d'IROISE (source SHOM).

2.1.2. Histoire

L'**I**nstitut **U**niversitaire **E**uropéen de la **M**er (**IUEM**) a été créé le 2 mai 1997 par le décret N°97-548 publié au Journal Officiel du 10 mai 1997. " École Interne " de l'Université de Bretagne Occidentale (UBO), l'**IUEM** est régi par l'article 33 de la loi des Universités de 1984. Son Directeur, M. le Professeur Paul Tréguer, est nommé par le Ministère de la Jeunesse, de l'Éducation et de la Recherche (MJER). L'**IUEM** est doté d'un Conseil Scientifique comprenant 26 membres dont 13

sont extérieurs à l'UBO (et originaires des principaux Instituts d'Amérique du Nord et d'Europe). L'**IUEM** accueille trois UMR de la Fédération de Recherche FR 2195 du CNRS, laquelle est rattachée au Département des Sciences de l'Univers.

En créant l'**IUEM**, l'Université de Bretagne Occidentale entend structurer son "Axe Mer", contribuer au développement du "Réseau Bleu" et à la restructuration du secteur environnement marin menée en Europe occidentale et septentrionale. L'**IUEM** est, depuis le 18 décembre 2000, membre du "Réseau National des Stations Marines" sous la houlette du CNRS. Il accueille plus de 500 personnes comprenant 150 permanents (dont environ 30% de personnels CNRS), une quarantaine de personnes sous contrats et environ 340 étudiants et doctorants.

2.1.3. Les Services Communs de l'IUEM

Les services communs de l'**IUEM** sont au service des équipes de recherche et d'enseignement :

- secrétariat et comptabilité,
- logistique embarquée et plongée scientifique,
- salles climatiques,
- pôle analytique (spécialisation en spectrométrie de masse),
- pôle de calcul intensif (en cours de création),
- service de documentation en liaison avec le Centre Européen de Documentation sur la Mer.

2.1.4. Organigramme

- | | |
|--|-------------------|
| > Direction de l' IUEM et FR 2195 : | Paul Tréguer |
| > Directeur-adjoint : | Pascal Tarits |
| > Direction de l'EDSM : | Jean Francheteau |
| > Responsable de l'Observatoire du Domaine Côtier : | Yannick Lageat |
| > Responsable des Services administratifs et communs : | Christine Estrade |

2.1.5. Quelques chiffres :

L'**IUEM** dispose de 1100 m² de bureaux, de laboratoires, et de salles d'enseignements, implantés sur un site unique face au goulet de la rade de Brest, et à proximité des autres centres de recherches et entreprises de la Technopôle Brest-Iroise.

(<http://www.tech-brest-iroise.fr>)

Des bureaux d'accueil sont à la disposition de visiteurs scientifiques qui bénéficient, lors de leurs séjours, des services communs de l'Institut.

Un ensemble unique d'équipements permet d'accueillir congrès et séminaires :

- amphithéâtre de 238 places,
- 7 salles de réunions attenantes de 30 à 80 places,
- 1 ensemble de visio-conférence.

2.1.6. Description de la cellule d'accueil : **Géomer UMR 6554**

Le laboratoire **Géomer** est né de la fusion des laboratoires Géosystèmes et Géolittomer à la fin de l'année 2002.

GEOMER (responsable : F. Gourmelon, Chargée de Recherche au CNRS), appartient à l'Unité Mixte de Recherche (UMR) 6554 du CNRS intitulée "Littoral, Environnement, Télédétection, Géomatique". Cette unité multisite, dirigée par Françoise Gourmelon, rassemble les quatre équipes suivantes : Costel-Rennes II, **GEOMER-Brest**, Géolittomer-Nantes et Géophen-Caen.

Cette UMR affiche clairement sa vocation pluridisciplinaire par l'origine de ses chercheurs (cartographie, écologie, géographie, géologie) et l'ouverture de ses collaborations aux champs disciplinaires des Sciences de l'Homme, de la Vie et de l'Univers, ainsi qu'aux Sciences physiques pour l'ingénieur. Elle est composée de 29 maîtres de conférence, de 17 professeurs, de 3 chargés de recherche, de 9 ingénieurs d'étude, d'un ingénieur de recherche, de 2 techniciennes et de 20 doctorants.

Le Laboratoire **GEOMER-Brest** a pour objectif d'analyser les systèmes littoraux à toutes les échelles spatio-temporelles en s'inscrivant dans une problématique située à l'intersection des champs de la géographie physique et de la géographie humaine. La double approche, naturaliste et sociale de l'environnement littoral, conduit à une réflexion concernant l'aménagement et la gestion intégrée des espaces côtiers dans une perspective de développement durable. Les principaux thèmes d'étude sont les suivants :

Gestion de la nature littorale : analyse des dynamiques naturelles des milieux littoraux (France de l'Ouest, Iles britanniques, Maroc, Gabon et Indonésie).

Géographie halieutique : analyse des systèmes halieutiques, de leurs mutations et de leurs blocages.

Tourisme littoral : analyse des activités touristiques et du développement de nouveaux produits (thalassothérapie, plaisance, golfs...).

Sociétés littorales : analyse des pratiques et comportements des usagers.

Le Laboratoire , qui est l'un des Centres de Compétences Thématiques en "Géomatique et Environnement" du Département des Sciences de l'Homme et de la Société du CNRS, a pour objectif l'analyse et la modélisation, au sein de Systèmes d'Information Géographique, de la dynamique des paysages côtiers et continentaux à partir de l'intégration de produits d'images satellitaires et de mesures collectées sur le terrain, et l'expression cartographique comme outil d'aide à la décision en matière d'aménagement et de protection.

Les principaux thèmes d'étude sont les suivants :

- Méthodologie de l'analyse spatiale de l'environnement au sein de Systèmes d'Information Géographique,
- Cartographie et synthèse de l'environnement,
- Évolution des systèmes micro-insulaires,
- Dynamiques spatiales des écosystèmes naturels et anthropisés des zones tropicales et semi-arides de l'Afrique de l'Ouest.

2.2. Les moyens informatiques

2.2.1. Matériels

- Stations de travail UNIX,
- PC équipé de Windows 98/NT4/2000.

2.2.2. Outils

- Bureautique classique,
- Navigateur internet : Netscape 4.7 à 7.1, Internet Explorer 5/6 (Microsoft),
- Base de données : Access2000 (Microsoft),
- Méthodes d'analyses et spécification : UML, Objecteering 5.2.2 (SoftTeam),
- Outils de développement : C++ Builder (Borland),
- Retouches d'images et cartographie : Illustrator 10 (Adobe), Photoshop 7.0 (Adobe),
- SIG : Arc/Info 8 Station (ESRI),
- Géoimage Station : Traitement d'images Satellitaires.

2.2.3. Le droit de l'informatique au laboratoire Géomer

Le laboratoire **Géomer** n'a pas de charte informatique ou de texte décrivant des droits et devoirs spécifiques au niveau de l'informatique, par contre le laboratoire est contractualisé avec l'UBO et il fait parti du CNRS, de part ces deux affiliations, les différents textes émis par l'UBO et le

CNRS s'appliquent également à **Géomer**. Vous pouvez consulter ces textes et travaux à l'annexe 6.

2.3. Le sujet de stage

Il s'agit de mettre en place sous forme informatique une "**Mémoire Environnementale**" de l'observation côtière dans la zone géographique de la Mer d'IROISE. Cette mémoire est composée elle-même de quatre mémoires : "*organisationnelle*", "*globale*", "*documentaire*" et "*d'observation*". La "*Mémoire Organisationnelle*" contient la description des acteurs et de l'organisation humaine de l'**IUEM** : ressources humaines, partenariats. La "**Mémoire Globale**" vise à dresser une "cartographie" synthétique des informations de l'**IUEM**, de l'expérience, des métiers et des compétences de ses acteurs (savoir-faire, pratiques météorologiques et modélisation). La "*Mémoire Documentaire*" contient les résultats, les publications issus des projets en conservant un lien avec ses auteurs et donc avec les acteurs. La "*Mémoire d'Observation*" contient des données d'observation issues des stations de mesure (comme celle de l'Observatoire du Domaine Côtier), des relevés, des cartes, des photos et des données de SIG ; elle contient également un catalogue qui est formé des métadonnées et prétraitement sur les données.

Le contenu de ces quatre mémoires doit être en partie consultable depuis l'extérieur de l'**IUEM** (internet) et en partie consultable seulement en intranet.

Les techniques de navigation entre ces différentes mémoires pourront être celles des liens hypertextes entre les pages html.

Parallèlement à cela, il est prévu de réaliser un moteur de recherche intervenant sur l'ensemble de la "**Mémoire Environnementale IROISE**". Ce moteur de recherche devra, dans la mesure du possible, sélectionner les documents en fonction de leur intérêt réel par rapport aux mots clés saisis.

Dans le cadre du stage, seules les mémoires "*organisationnelle*" et "*documentaire*" seront étudiées afin de construire une première ébauche du système.

Ceci inclus :

- la conceptualisation des projets de l'**IUEM** en rapport avec le domaine côtier de la Mer d'IROISE, la conceptualisation des participants à ces projets et la documentation qui leurs sont associées.
- la construction d'une base de données
- l'implémentation de quelques cas d'utilisations pour valider certains choix structuraux.

3. PRESENTATION DU TRAVAIL



Cette nouvelle partie présente le travail effectué tout au long de la période de stage. La première étape étant l'analyse du champ de l'étude et du domaine, dans un deuxième temps l'analyse des besoins pour le nouveau système, puis l'implémentation de la base de données et de cas d'utilisations remarquables.

3.1. Analyse du champ de l'étude

L'analyse du champ de l'étude permet de définir la base du projet. Elle détermine les limites, les objectifs et les enjeux du projet.

3.1.1. Description des tâches

Les objectifs souhaités sont de créer une bibliothèque de l'environnement grâce aux données collectées dans la région de la Mer d'Iroise et également de suivre l'évolution des projets de l'**IUEM**, ce vaste chantier est programmé sur deux années et scindé en quatre étapes :

Extrait de la réponse de l'appel à participation (voir annexe 2) :

«

Etape 1 : (durée de réalisation 6 à 8 mois)

Modélisation conceptuelle de la mémoire environnementale "Iroise" : cette étape nécessitera une démarche en deux temps : dans un premier temps, il sera nécessaire de dresser le schéma systémique de l'observatoire et de ses acteurs, des questions scientifiques fondatrices, des séries de données et de leurs relations. Ce schéma de système, à la réalisation duquel seront déjà associés les spécialistes du traitement de l'information, sera ensuite utilisé pour développer une modélisation UML décrivant la mémoire environnementale et ses interconnexions.

Etape 2 : (durée de réalisation 12 mois)

Inventaire de l'ensemble des composantes du système. Le but de cette étape est d'inventorier les éléments de chacune des quatre mémoires du système d'observation, de mesurer le niveau d'hétérogénéité, d'évaluer le travail nécessaire pour normaliser les différents gisements sous des formes exploitables.

On peut noter que la partie "*Mémoire d'Observation*" est celle qui demandera le moins de travail car le système d'information géographique de l'observatoire est déjà bien avancé. Une problématique méthodologique sera cependant abordée : le stockage des grandes masses de données-image et leur pérennisation.

Etape 3 : (durée de réalisation 6 à 8 mois)

Sélection/développement et test des logiciels de gestion, stockage et analyse des quatre types de mémoires. Analyse des capacités d'interopérabilité des catalogues de données selon le modèle conceptuel défini à l'étape 1, sélection des outils de développement des bases de connaissances, réflexion sur la prise en compte du temps dans l'analyse et la restitution des contenus de la mémoire environnementale.

Etape 4 : (démarrage au début de la 2^{ème} année; durée de réalisation 12 mois)

Mise en place d'un pilote de la mémoire environnementale "Iroise". Ce pilote intégrera un modèle conceptuel ouvert et évolutif, intégrant la description des relations et des contenus des quatre sous-mémoires. Pour chacune d'entre elles, tout ou partie¹ des données existantes sera catalogué et stocké sous une forme mobilisable.

Un prototype d'ERP (à définir) sera mis en œuvre pour répondre aux besoins de l'exploitation de cette mémoire environnementale.

»»

Mon intervention dans le cadre de ce projet se porte principalement sur la première phase et dans une moindre mesure sur les deuxième et troisième phases. Après avoir revu les objectifs, nous avons pu cibler les tâches sur lesquelles nous pourrions raisonnablement travailler pendant la durée du stage. C'est à dire :

- la conceptualisation de l'archivage de l'organisation et de la documentation de l'**TUEM** autour de ses projets, ce qui constitue l'analyse du modèle du domaine pour laquelle nous avons créé un package spécifique intitulé : « **mod domaine** ».
- La réalisation d'une maquette du système permettant de valider une partie de la conceptualisation, notamment en réalisant l'implémentation de cas d'utilisations faisant intervenir la notion d'archivage, ce qui constitue l'analyse du modèle d'application à laquelle nous avons aussi consacré un package intitulé : "**mod appli**".

L'étude peut donc se diviser en deux parties, une pour chaque package, une consacrée à l'étude du domaine et l'autre à l'étude de l'application. En revanche ce découpage n'implique pas une dissociation des deux, car il subsiste une dépendance du modèle de l'application envers le modèle du domaine. Un exemple simple justifie cette dépendance que nous pouvons qualifier « d'ordre de traitement », cet exemple est celui de la base de données qui ne peut être schématisée que si le modèle de données l'est lui aussi.

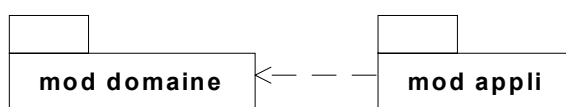


Figure 2 : Schéma des packages projets.

Répartition de mon intervention dans le projet au cours de mon stage :

	Phase 1 : conceptualisation	Phase 2 : capture des données	Phase 3 : implémentation
Spécification/conceptualisation du système pour les Mémoires Organisationnelles et Documentaires	X	X	
Implémentation d'une maquette			X

Estimation de la répartition horaire :

Phase	1	2	3
%	65	5	30
Temps à passer (h)	500	40	230

Pour le package **mod domaine** deux types de tâches se dégagent, une première doit permettre d'identifier les principaux objets du système, ceci sera réalisé lors de la modélisation statique du système. Une deuxième tâche doit permettre de capter les attentes des utilisateurs envers le nouveau système, c'est la donc la phase de capture des besoins fonctionnels et de la définition des cas d'utilisations (CU).

L'étude du package **mod appli** va constituer le début de la phase de prototypage avec la schématisation de la base de données, puis son implémentation et enfin l'implémentation des premiers cas d'utilisations. Nous devons organiser ces différentes tâches dans le temps, ainsi qu'estimer le temps imparti pour chacune d'entre elles. Pour cela deux plannings prévisionnels ont été créés (voir annexe 5), l'un commence dès le début du stage, l'autre débute à partir de la semaine 27. Ce deuxième planning s'est avéré nécessaire afin de redistribuer les tâches dans le temps après les dix premières semaines de stage.

3.2. Analyse du domaine

Nous allons maintenant définir le domaine de l'application et décrire les objets qui la composent. Cette étude est restreinte à l'analyse des mémoires dites "*organisationnelle*" et "*documentaire*" comme nous l'avons vu précédemment. La "*Mémoire d'Observation*" sera seulement abordée et intégrée aux diagrammes de classes pour montrer ce qu'elle pourrait être.

3.2.1. Définition et limites du domaine

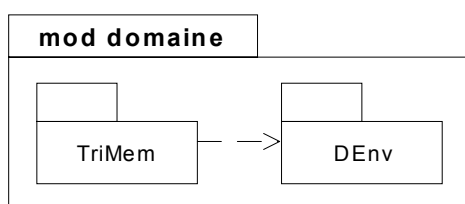
Après l'étude du cahier des charges et un début de conceptualisation, il apparaît que quatre objets se distinguent dans le domaine à spécifier : ce sont les acteurs, les projets, les productions et les données. Ces objets sont à l'intérieur d'un cadre qui est l'**IUEM**.

Tout d'abord définissons l'**IUEM** : l'Institut Universitaire Européen de la Mer qui est à la fois un centre de recherche et un centre de formation universitaire qui accueille des étudiants à partir de la Licence jusqu'au Doctorat. Le système à mettre en place, **MEnIr**, sera tout d'abord réservé aux personnes faisant parti de ce même **IUEM**. D'où les futurs utilisateurs seront, des étudiants, le personnel des différents laboratoires et le personnel commun de l'**IUEM**.

- *les projets* : sont ceux qui naissent des activités de recherche présentes à l'**IUEM** et dont la responsabilité est assurée par une personne de l'**IUEM**.
- *les acteurs de la mémoire "organisationnelle"* : les personnes présentes à l'**IUEM** (celles citées précédemment), ainsi que celles externes à l'**IUEM** et qui participent aux projets.
- *les productions* : elles peuvent être sous plusieurs formes : papiers, numériques et de différents types : publications, livres, thèses, rapports, cartes, ... Nous souhaitons archiver les productions produites à partir d'un projet et également celles utilisées lors d'un projet.
- *les données de la mémoire d'observation* : elles proviennent des stations de mesures qui composent l'Observatoire du Domaine Côtier, certaines bases de données de Météo France, le trait de côte du SHOM, des données de l'IGN, ...

Une caractéristique essentielle de **MEnIr** est la notion "d'archivage", celle-ci est valable pour les quatre mémoires.

3.2.2. La décomposition en packages d'étude pour "mod domaine"



Afin de faciliter l'étude et de regrouper les entités, deux packages ont été créés à l'intérieur du modèle du domaine : TriMem et DEnv. TriMem pour les mémoires "*organisationnelle*", "*documentaire*" et "*globale*" et DEnv pour la mémoire "*d'observation*".


Figure 3 : Schéma des packages du modèle du domaine.

Le lien de dépendance entre les deux packages s'explique par notre souhait de rendre la "*Mémoire d'Observation*" indépendante des autres mémoires, afin qu'elle puisse être facilement exportable et réutilisable.

Cette décomposition va également nous aider au niveau de la gestion du projet, en effet ce découpage est en adéquation avec les objectifs fixés pour la période de mon stage : c'est-à-dire l'étude des mémoires de TriMem.

3.2.3. Définition des acteurs

Les acteurs du système sont ici les personnes physiques qui vont interagir avec le système d'information, autrement dit se sont les utilisateurs du système.

	Iuemien(ne)	Toute personne qui fait parti de l' IUEM .
	Auteur	Personne qui produit une production et qui est cité comme auteur dans celle-ci.
	Responsable de projet	Personne responsable ou co-responsable d'un projet.
	Administrateur	Administrateur du système, il gère le système, met à jour et s'assure de l'intégrité : - des données insérées dans la/les base(s) de données - des informations diffusées par le système.

3.2.4. Construction et description des diagrammes de classes

... plusieurs niveaux de modélisations coexistent ...

- **1^{er} niveau** > Modélisation de l'archivage :

L'archivage est la clé de voûte de la "**Mémoire Environnementale IROISE**". Comme son nom l'indique, nous souhaitons que ce système puisse constituer une mémoire de l'environnement dans la zone géographique de la Mer d'IROISE. Pour cela un effort de conceptualisation est nécessaire :

1. tout d'abord nous devons identifier les objets qui doivent être archivés,
2. puis nous devons trouver la ou les structures adéquates afin d'assurer la pérennité et l'accessibilité de ces données.

Après discussions, réflexions et quelques sondages, il apparaît que trois objets sollicitent une structure d'archivage : « acteur », "projet" et "production". Deux cas s'offrent à nous :

- le *premier* concerne l'objet « acteur » qui nécessite une organisation particulière, car les modifications apportées sur une partie de ces attributs ne doivent pas être conservées, comme le nom, le prénom, les numéros de téléphone et fax, le mail, ... L'ancienne valeur de ces attributs est inutilisable, seule la valeur actuelle l'est, un exemple simple est celui du mail : un mail ne sert à rien du moment qu'il n'existe plus et ne pourra plus servir à l'avenir donc on peut le remplacer par sa nouvelle valeur sans l'archiver.

Liste des attributs >>
Les modifications sur tous les attributs contenus dans la classe « acteur » ne nécessitent pas d'être archivées.

acteur
id_acteur : integer
nom_acteur : string
prenom_acteur : string
num_tel_acteur [0..1] : string
mail_acteur [0..1] : string
num_fax_acteur [0..1] : string
type_metier [1..n] : typeMetier
sexe : string
url_site_perso_acteur : string
url_photo_acteur : string

En revanche l'archivage est nécessaire pour ce qui concerne les compétences acquises par un acteur, ainsi que dans le cas d'un changement de statut ou d'équipe de rattachement. Ces informations, qui apparaissent dans l'historique de l'acteur et lors de la consultation d'un projet, nous sont utiles pour avoir plus de renseignements sur le contexte et la position d'un acteur dans un projet à un instant donné. Par exemple : au 20/10/2001, Monsieur X fait parti de l'entreprise Y et travaille sur le projet P1. Le 18/02/2002, il change de société et il représente l'entreprise W et il travaille toujours sur le projet P1. Il serait peut-être intéressant de voir plus tard l'influence de ce changement sur le projet. Un autre exemple, un doctorant devenu maître de conférence. Pour l'étude de l'évolution du projet, il s'avère utile de savoir que cette personne a participé au projet comme doctorant à ses débuts et qu'elle l'a poursuivi comme maître de conférence.

- le *deuxième* cas porte sur les objets « projet » et « production » : contrairement à la classe acteur, les attributs des classes projet et production doivent être archivés.

Pour les projets, nous souhaitons pouvoir créer des périodes de projet qui les unes accolées aux autres vont former une chaîne de périodes. En fait, une période de projet est une nouvelle instance de projet dans laquelle un attribut va conserver l'identifiant du projet précédent. Voilà les périodes de projet créées et ce sont-elles qui permettent d'archiver les attributs de la classe projet, concrètement si le contexte est réécrit et qu'il est indispensable de conserver l'ancienne version, alors il suffira de préciser au système de créer une nouvelle période, donc une nouvelle instance du projet, qui aura pour "contexte" la nouvelle version. L'ancienne version pourra être consultée via l'historique du projet.

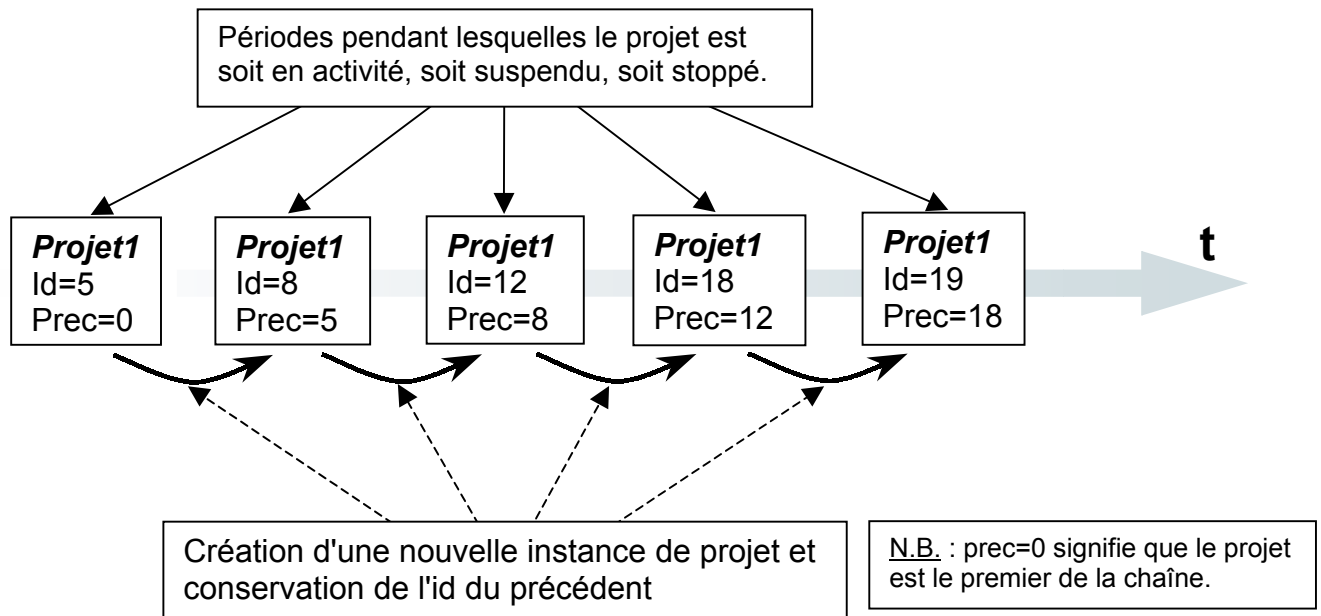


Figure 4 : Chaînage des projets.

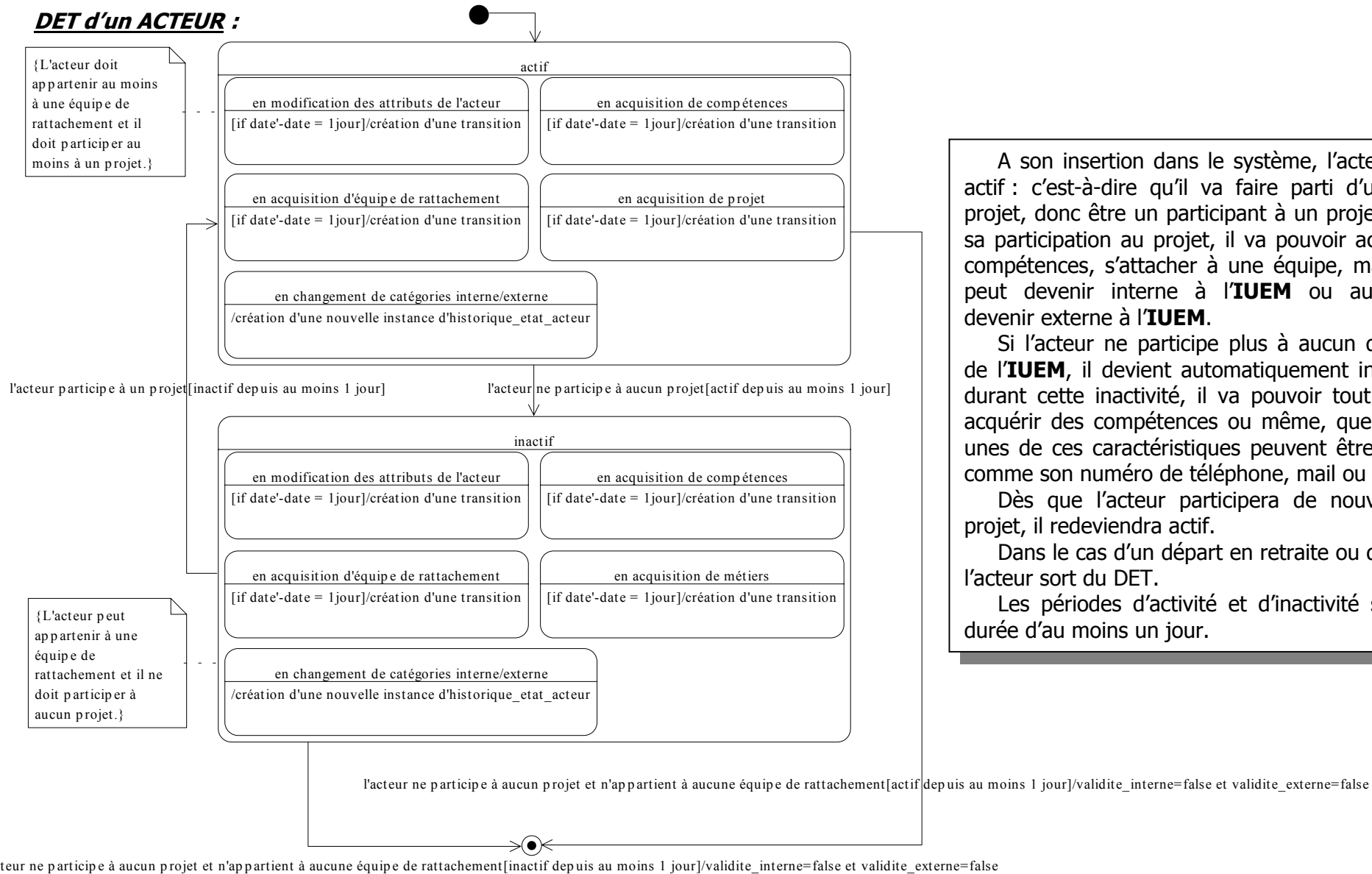
En ce qui concerne les productions, le même système sera mis en place. L'utilisation de ce système sera très intéressante pour le suivi des plannings et des normes par exemple.

- **2^{ème} niveau** > Modélisation des objets acteurs, projets et productions :

Afin de déterminer les attributs nécessaires aux objets intégrés dans la "**Mémoire Environnementale**", nous avons rédigé un questionnaire destiné aux futurs utilisateurs du système. Ce questionnaire a été diffusé par mail, puis nous sommes allés interviewer les personnes pour recueillir leurs souhaits et leurs attentes d'un tel système. Ce questionnaire est constitué de deux parties, une pour la présentation du projet et l'autre pour les questions (voir annexe 2).

Grâce à ceci, nous avons pu déterminer la structure et les différents états de chaque objet, ce qui nous amène aux diagrammes d'états-transitions (DET) qui suivent.

DET d'un ACTEUR :



A son insertion dans le système, l'acteur va être actif : c'est-à-dire qu'il va faire parti d'une équipe projet, donc être un participant à un projet. Pendant sa participation au projet, il va pouvoir acquérir des compétences, s'attacher à une équipe, mais aussi il peut devenir interne à l'**IUEM** ou au contraire devenir externe à l'**IUEM**.

Si l'acteur ne participe plus à aucun des projets de l'**IUEM**, il devient automatiquement inactif. Mais durant cette inactivité, il va pouvoir tout de même acquérir des compétences ou même, quelques-unes de ces caractéristiques peuvent être modifiées comme son numéro de téléphone, mail ou autres.

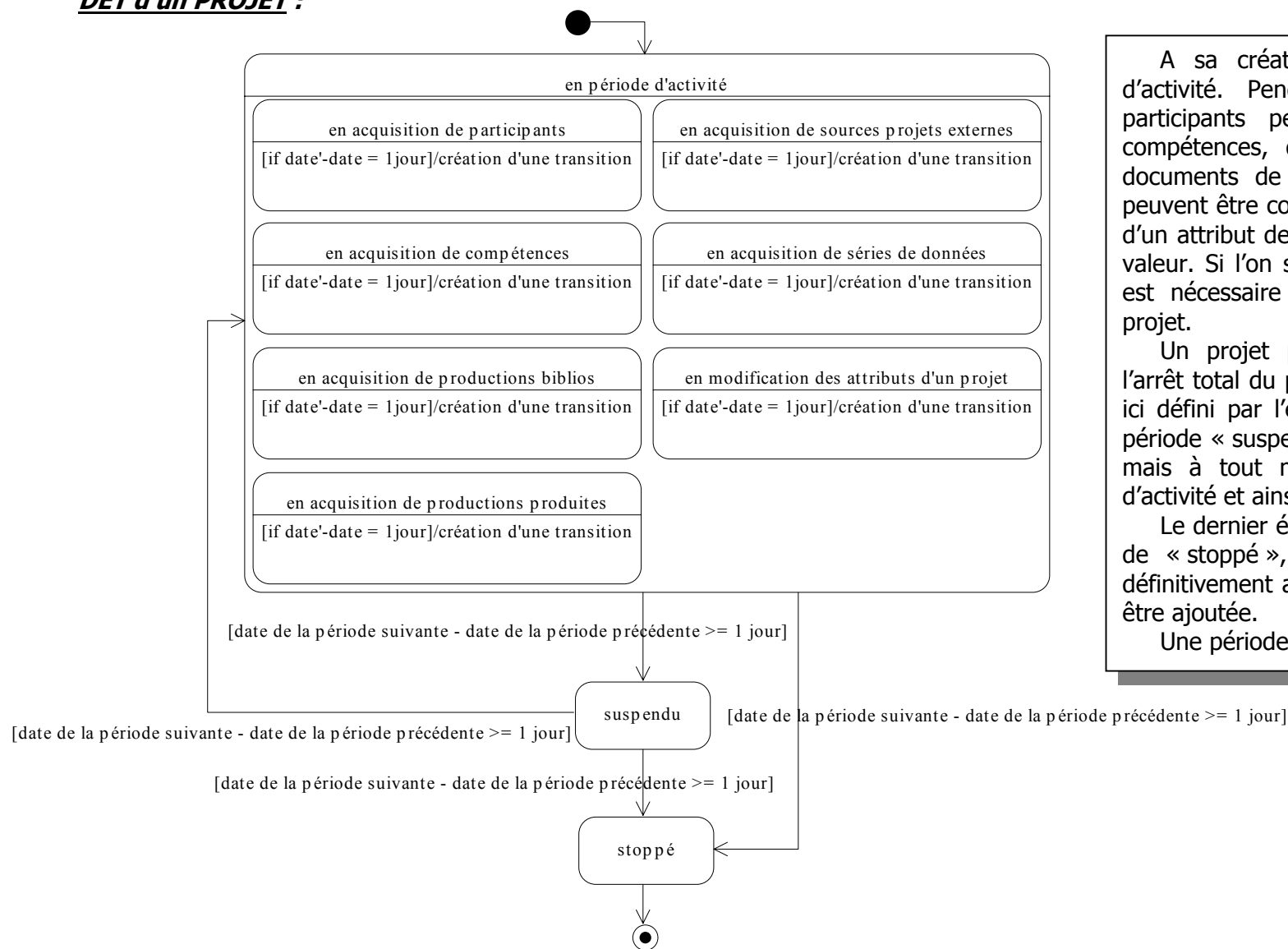
Dès que l'acteur participera de nouveau à un projet, il redeviendra actif.

Dans le cas d'un départ en retraite ou d'un décès, l'acteur sort du DET.

Les périodes d'activité et d'inactivité sont d'une durée d'au moins un jour.

Figure 5 : DET d'un acteur.

DET d'un PROJET :



A sa création le projet va être en période d'activité. Pendant celle-ci, la composition des participants peut être modifiée, les listes des compétences, des productions, des données, des documents de suivi de projet associés au projet peuvent être complétées. A noter que la modification d'un attribut de projet annule et remplace l'ancienne valeur. Si l'on souhaite archiver ces modifications, il est nécessaire de créer une nouvelle période de projet.

Un projet peut ne plus avoir d'activité, mais l'arrêt total du projet n'est pas envisagé, cet état est ici défini par l'état dit « suspendu », pendant cette période « suspendu », le projet ne peut pas évoluer, mais à tout moment il peut revenir en période d'activité et ainsi à un cours normal.

Le dernier état possible pour un projet, c'est celui de « stoppé », dans ce cas précis le projet est définitivement arrêté, plus aucune période ne pourra être ajoutée.

Une période de projet dure au moins un jour.

Figure 6 : DET d'un projet.

- Quelques particularités de modélisation : l'utilisation des "types", des "types énumérés" et de "struct".

Types : "Ensemble d'attributs, de relations et d'opérations qui précisent une structure réalisée par une classe ou un composant. Nous avons utilisé ceci dans le cas du chaînage des projets : un projet possède un attribut qui fait référence à un autre projet, par conséquent le type de l'attribut sera "projet", au lieu "d'integer", ceci permet d'introduire la notion d'objet et d'avoir un meilleur niveau de précision pour le type référencé".

Struct : "Liste d'attributs connexes (structure de données), utilisée comme type d'attribut".

Les classes "publication" et "rapport" n'ont pas la même origine, par conséquent elles ne peuvent pas hériter classiquement des attributs qu'elles ont en commun. Ainsi nous avons utilisé une classe intitulée "**refBiblio**" et stéréotypée "struct", pour regrouper ces attributs.

Types énumérés : "Liste de valeurs valides pour un attribut d'une classe".

Le "type énuméré" équivaut à une classe qui possède deux attributs : un identifiant et un attribut qui permet de conserver une valeur, dans notre cas ce sont principalement des chaînes de caractères (string). Ces valeurs sont prédéfinies et peuvent même faire l'objet d'une contrainte de type OCL (voir figure 7). Son utilisation permet d'alléger les diagrammes de classes. En revanche, la structure de la base donnée est inchangée, sauf dans le cas où le SGBD utilisé permet de gérer des énumérations.

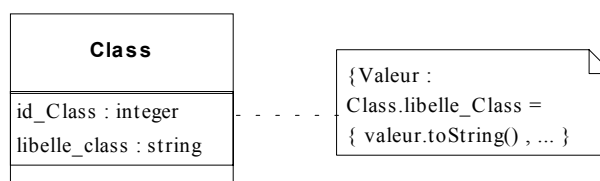


Figure 7 : Equivalence d'une classe de type énuméré.

Après toutes ces démarches d'enquête et de réflexion sur la structure du domaine étudié, il nous a été possible de fabriquer un modèle contenant à la fois une structure représentative de l'organisation (acteurs) de l'**TUEM** et une structure représentative de la documentation (productions) associée aux projets, tout en assurant l'archivage et la conservation des relations existantes entre ces deux structures. Le résultat de cette analyse est composé des sept diagrammes de classes suivants.

La rubrique "*Diagrammes de classes*" de l'annexe 1 contient trois diagrammes : un qui englobe l'ensemble de la "*Mémoire Environnementale*", un autre spécialisé dans la description du modèle de la "*Mémoire Organisationnelle*" et le troisième qui décrit la "Mémoire Documentaire". Nous avons préféré éclater ces trois diagrammes en sept pour faciliter la lisibilité et pour laisser apparaître les classes prépondérantes.

Nous allons tout d'abord aborder les Diagrammes de classes de la "*Mémoire Organisationnelle*". L'ensemble de la "*Mémoire Organisationnelle*" est représenté par trois diagrammes (figures 8 à 10).

Au sein des acteurs de l'**IUEM** on distingue deux grandes catégories : les acteurs internes et externes à l'**IUEM**. Les acteurs externes à l'**IUEM** peuvent être fournisseur externe de données. Les fournisseurs externes sont soit conventionnés, soit non conventionnés suivant les accords passés avec l'**IUEM**.

Ces différentes personnes peuvent travailler sur plusieurs projets et occupées une place de responsable ou coresponsable pour ces projets. Les acteurs possèdent des caractéristiques propres telles que leur nom, prénom, numéro de téléphone, etc., mais aussi des métiers

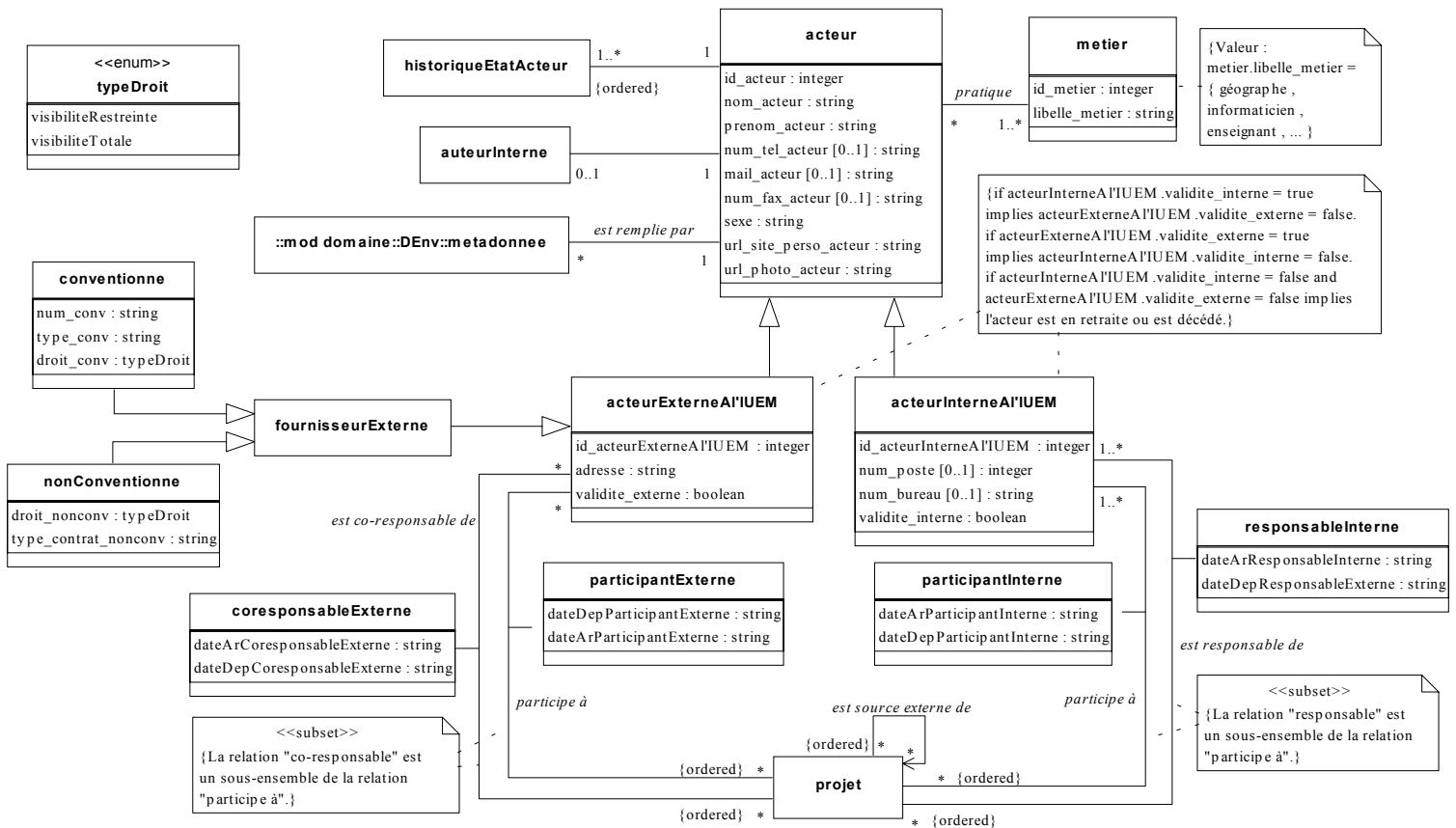


Figure 8 – Diagramme de classes décrivant la structure acteur.

Une équipe de rattachement est une entité à laquelle est attachée des personnes, elle peut être un laboratoire, un institut, une entreprise, une unité, ... Celle-ci possède une ou plusieurs provenance(s) selon la répartition géographique des membres de l'équipe.

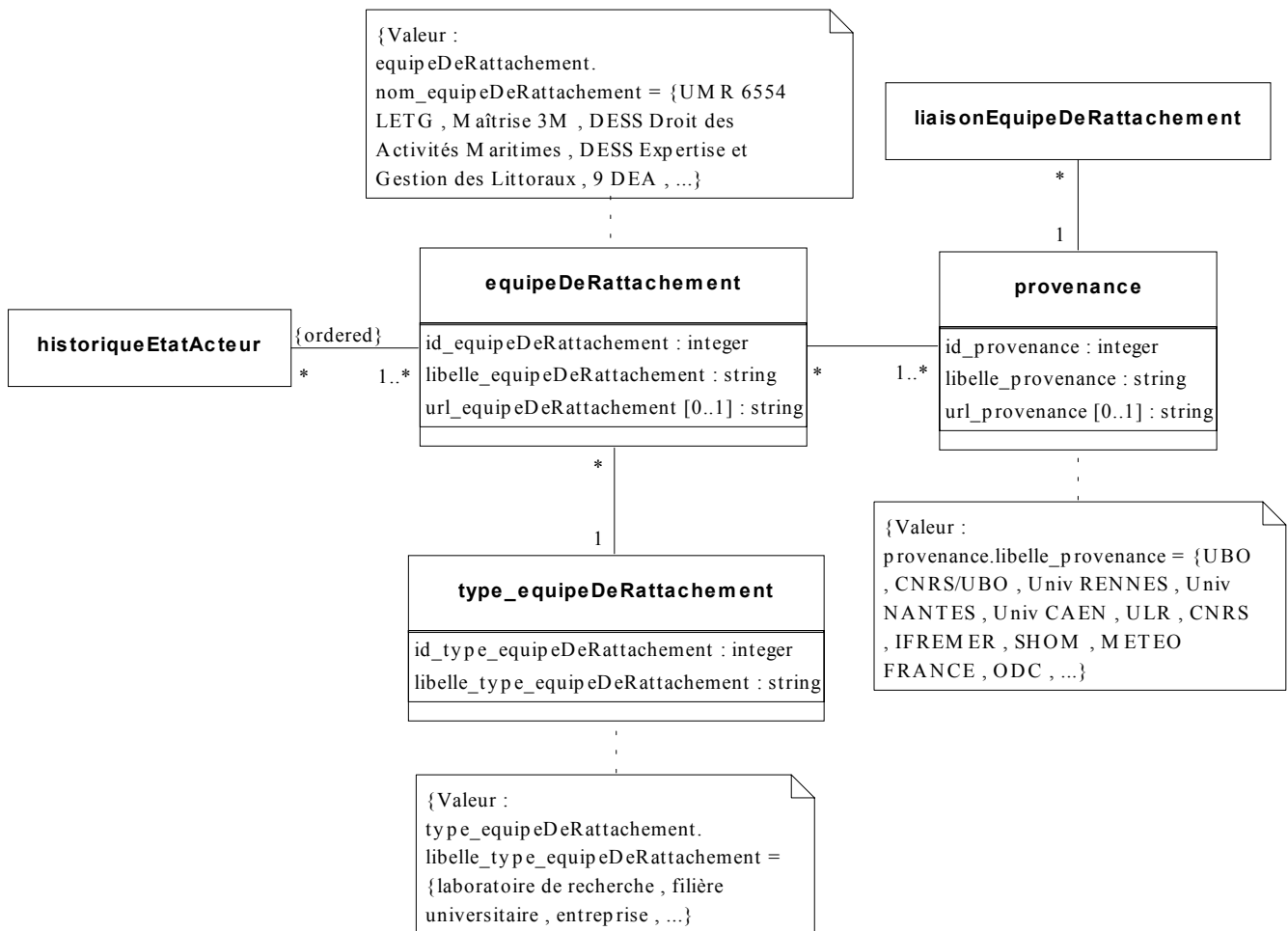


Figure 9 : Elément "Equipe de rattachement".

La réflexion menée sur l'archivage des acteurs a débouché sur deux conclusions :

- les attributs de la classe "acteur" ne doivent pas être archivés,
- l'ajout de compétences et d'équipe de rattachement doit être archivé pour permettre la constitution d'un "historique du parcours" d'un acteur. Cet archivage est réalisé à l'aide de la classe "historiqueEtatActeur" qui recueille, à partir d'une date, l'état de l'acteur et elle va servir de base pour connecter les compétences et les équipes de rattachement à une date donnée. C'est-à-dire que l'acteur va avoir des périodes qui vont se juxtaposer, ces périodes possèdent un état d'acteur, qui est soit actif, soit inactif. Pendant ces périodes, il va donc être possible d'ajouter des compétences et des équipes de rattachement.

A chaque ajout d'une équipe de rattachement à une instance d'historiqueEtatActeur, un statut et une provenance doivent leur être associés pour connaître, quel statut a la personne dans tel ou tel équipe et où travaille-t-elle.

A noter : les compétences sont de différents types, comme par exemple : des compétences de modélisation, des pratiques métrologiques ou des savoir-faire.

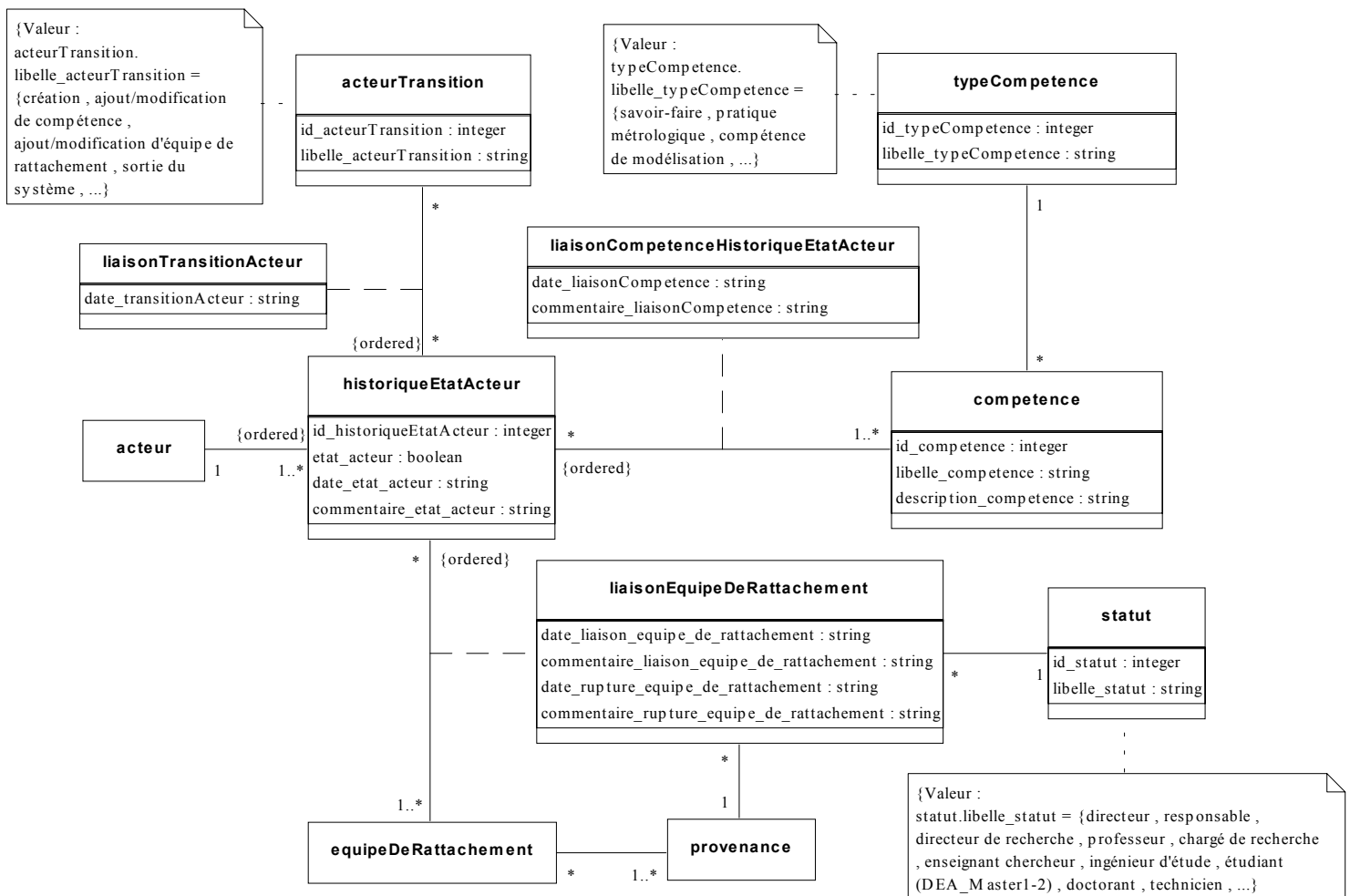


Figure 10 – Diagramme de classes décrivant la structure de l'historique pour un acteur

Après avoir vu la structure de la "*Mémoire Organisationnelle*", nous allons aborder maintenant les Diagrammes de classes de la "*Mémoire Documentaire*", ils sont au nombre de quatre (figures 11 à 14).

La documentation se divise en plusieurs parties, voici l'une d'entre elles : le "questionnement scientifique".

Le questionnement scientifique hérite de documentation et il est associé à une ou plusieurs thématique(s).

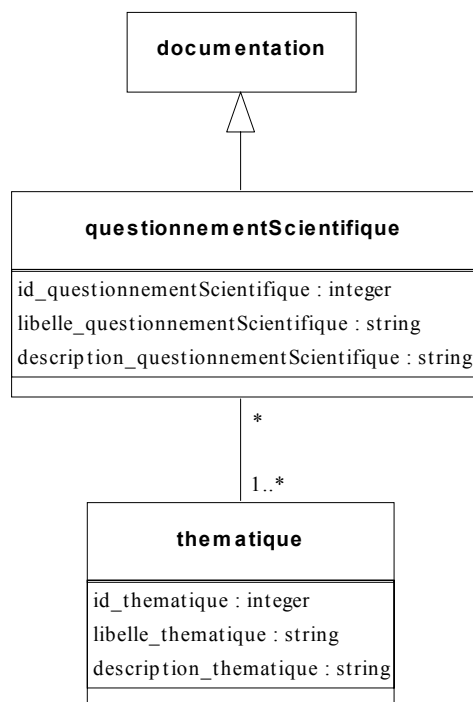


Figure 11 - Diagramme de classes décrivant le questionnement scientifique.

La "mesure" constitue une deuxième partie. Elle est composée des protocoles et des catalogues de données de la "Mémoire d'Observation" (package Denv). Le début de la spécification de la "Mémoire d'Observation" est représenté, ici, pour nous permettre d'imaginer sa future structure, mais ceci ne préfigure en rien la solution qui sera adoptée.

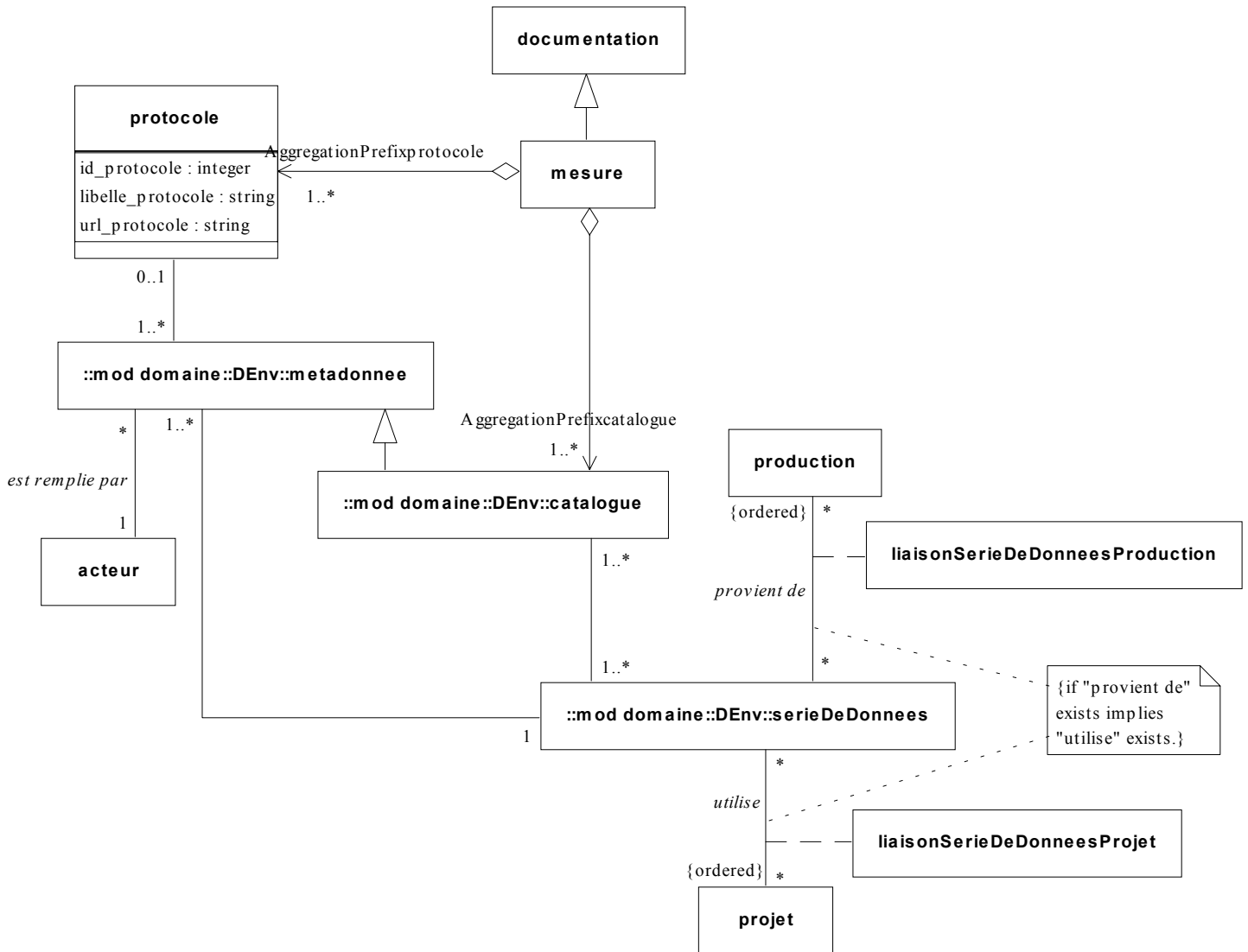


Figure 12 - Diagramme de classes décrivant la structure de la mesure.

Un projet possède des participants qui proviennent des acteurs internes ou externes à l'**IUEM**, au moins un des acteurs internes doit être responsable du projet et des acteurs externes peuvent lui venir en aide en tant que coresponsables.

Afin de suivre l'évolution d'un projet dans le temps, ils sont liés les uns aux autres, ainsi il est possible de constituer un historique du projet. Cette liaison s'effectue à l'aide de l'attribut "PROJET_SOURCE_PROPRE" qui a pour valeur les identifiants des projets. Contrairement aux acteurs, les attributs de la classe "projet" doivent être archivés. Par conséquent, à chaque fois s'il est souhaitable d'archiver une modification en conservant l'ancienne et la nouvelle valeur, il suffit de créer un nouvel état pour le projet. Il est possible de relier des données environnementales à un projet, ceci pour symboliser l'utilisation d'une donnée dans un projet, de même pour les productions (constitution d'une bibliographie). De plus au cours de la réalisation du projet des productions sont produites.

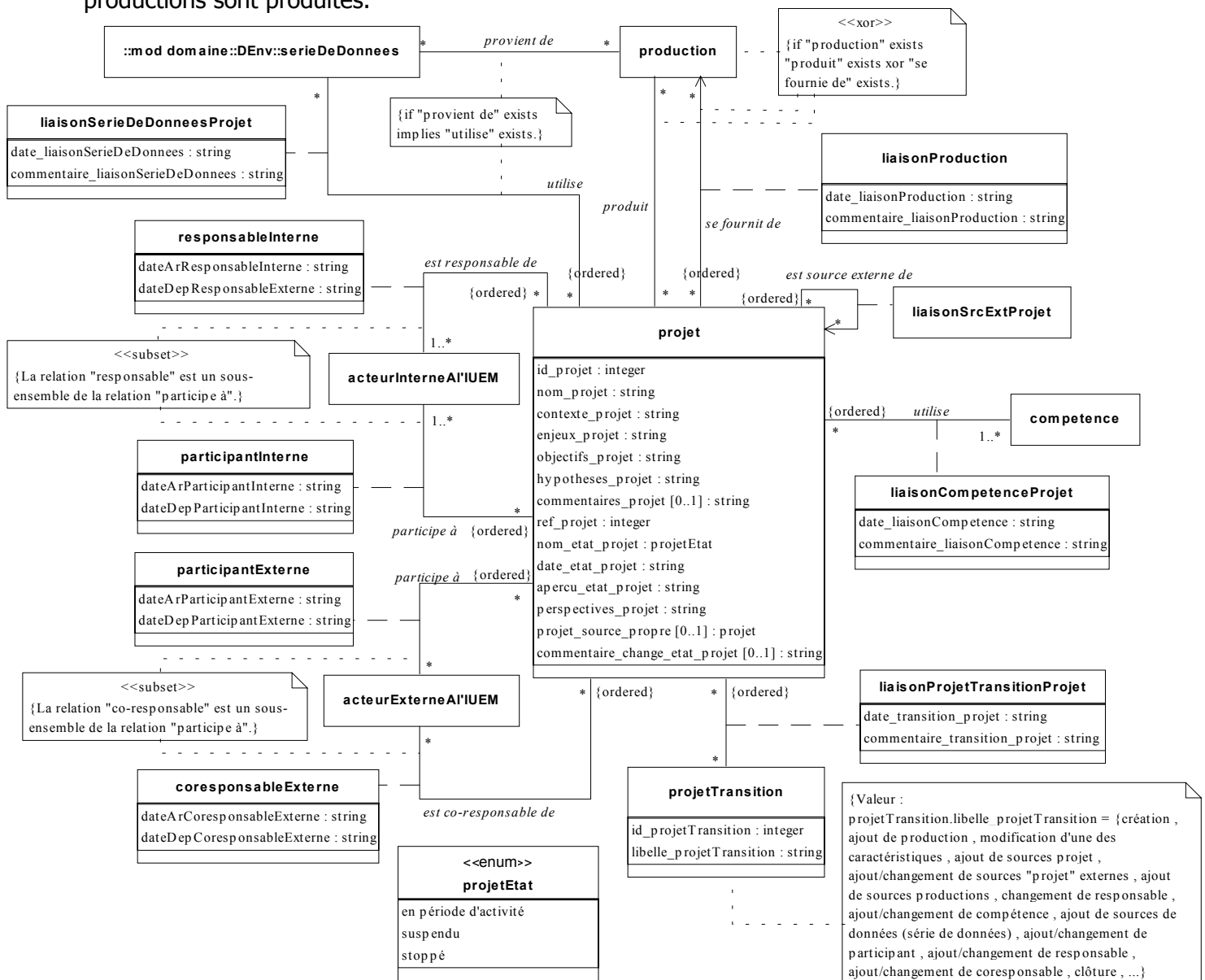


Figure 13 – Diagramme de classes de la Mémoire Documentaire : Projet.

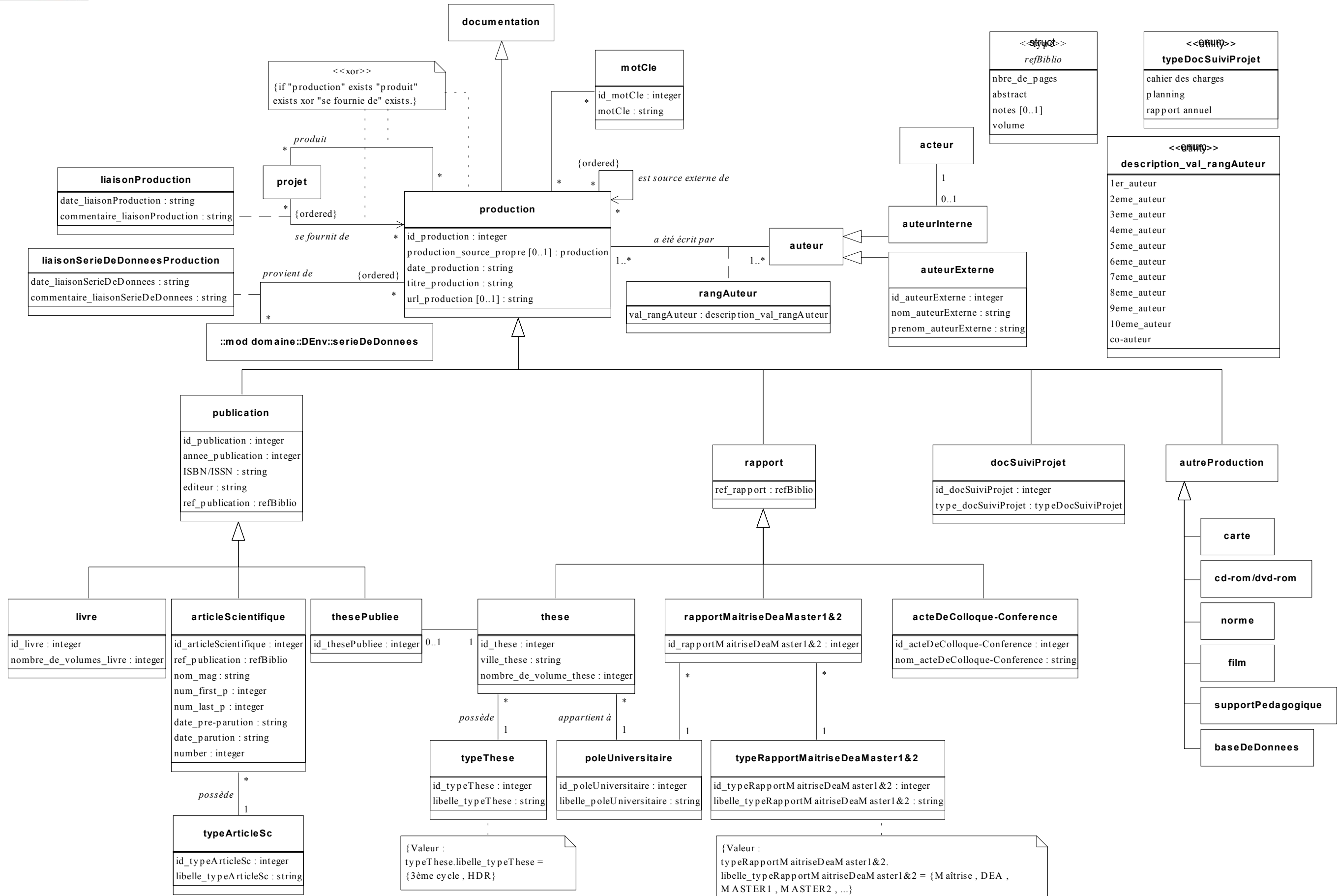


Figure 14 –Diagramme de classes de la Mémoire Documentaire : Production.

Les productions se trouvent sous plusieurs formes : des publications, des rapports, de la documentation de suivi de projet et d'autres productions. Les publications sont elles-mêmes divisées en articles scientifiques, thèses publiées, livres. Les rapports sont les thèses, rapports (Maîtrise, DEA, Master 1&2) et actes de colloque/conférence. Les autres publications correspondent aux cartes, normes, cd-rom/dvd-rom, films, supports pédagogiques, les bases de données, etc.

Pour assurer le suivi de l'élaboration des productions et des liens qui existent entre elles, celles-ci sont liées les unes, ceci se traduira par une table de relation en base de données et cette table de relation devra posséder deux références de l'identifiant de la table "production". Cet artifice va nous permettre de créer une bibliographie. Ces productions sont produites par des auteurs qui font partis des acteurs ou des auteurs externes, notamment pour les documents utilisés dans un projet, mais rédigé par une personne extérieure. Les auteurs sont classés par ordre à l'aide de la classe d'association "rangAuteur".

Vous pourrez trouver en annexe 1 le diagramme de classes général. Celui-ci est la réunification des diagrammes précédents, il permet de voir la schématisation du domaine dans sa globalité, mais aussi d'appréhender plus facilement les liens étroits entre l'organisation et la documentation, c'est-à-dire entre les acteurs et les projets, et entre les acteurs et les productions.

Pour faciliter la lisibilité, les attributs des classes n'apparaissent pas dans ce diagramme, pour consulter les attributs veuillez vous reporter aux diagrammes précédents.

3.2.5. Extrait du dictionnaire de données

Le dictionnaire de données permet de définir le sens des classes et de leurs attributs à l'aide d'une phrase ou d'un texte. Ce qui suit est un extrait du dictionnaire, car il ne nous semblait pas nécessaire de tout définir dans le rapport, par contre vous pouvez retrouver le dictionnaire complet à l'annexe 1.

Format du dictionnaire de données

Classe "<nom de la classe>"

Description

Texte descriptif de la classe

Attributs :

<portée> <type> <nom de l'attribut> : Texte descriptif de l'attribut.

Classe "projet"

Description

Cette classe rassemble tous les éléments qui définissent un projet. Un projet est constitué d'acteur qui sont soit internes, soit externes à l'**IUEM**. Les acteurs internes participent à un projet et peuvent être responsables. Les acteurs externes participent à un projet et peuvent être co-responsable. Par conséquent un projet est toujours dirigé par au moins un acteur interne, qui peut s'appuyer sur au moins un acteur externe qui occupe une place de co-responsable.

Types externes :

- *projet*

Attributs :

- *public integer id_projet* : Identifiant numérique permettant de référencer un projet.
- *public string nom_projet* : Nom du projet.
- *public string contexte_projet* : Contexte du projet.
- *public string enjeux_projet* : Enjeux du projet.
- *public string objectifs_projet* : Objectifs du projet.
- *public string hypotheses_projet* : Hypothèses du projet.
- *public Set(0..1) string commentaires_projet* : Commentaires sur le suivi du projet.
- *public integer ref_projet* : Le numéro/référence du projet, est la référence donnée par l'administration du projet.
- *public projetEtat nom_etat_projet* : Nom de l'état courant du projet.
- *public string[8] date_etat_projet* : Date de création du projet.
- *public string apercu_etat_projet* : Courte phrase, résumé sur le projet pour donner une indication, une vue succincte sur le projet, comme : "début du projet", "phase de récoltes de données", "phase d'exploitation de données". C'est cet aperçu qui va apparaître dans « l'explorer de projet » en plus du nom.
- *public string perspectives_projet* : Perspectives pour le projet, projections dans le futur.
- *public Set(0..1) projet projet_source_propre* : La source d'un projet est l'attribut qui permet de chaîner les projets entre eux pour conserver l'historique du projet.
- *public Set(0..1) string commentaire_change_etat_projet* : Commentaire sur l'origine du changement d'état du projet.

Classe "acteur"

Description

Classe supérieure de la définition d'un acteur. Elle contient le nom, le prénom de l'acteur et différents renseignements sur lui.

Attributs :

- *public integer id_acteur* : Identifiant numérique permettant de référencer un acteur.
- *public string[50] nom_acteur* : Nom de l'acteur.
- *public string[50] prenom_acteur* : Prénom de l'acteur.
- *public Set(0..1) string num_tel_acteur* : Numéro de téléphone de l'acteur.
- *public Set(0..1) string mail_acteur* : Adresse mail de l'acteur.
- *public Set(0..1) string num_fax_acteur* : Numéro de fax de l'acteur.
- *public Set(1..n) typeMetier type_metier* : Nom d'un (des) métier(s) de l'acteur.
- *public string sexe* : Sexe de l'acteur : homme/femme.
- *public string url_site_perso_acteur* : Lien vers le site internet personnel de l'acteur.
- *public string url_photo_acteur* : Lien vers la photo de l'acteur.

Classe "acteurExterneAl'IUEM"

Description

Les acteurs externes à l'**IUEM** sont un type d'acteur qui ne font pas partis de l'**IUEM** de Plouzané. Ils peuvent appartenir aussi bien à des entreprises, des instituts, ... ou bien être des membres d'une équipe de recherche.

Classes parentes :

- *acteur*

Attributs :

- *public integer id_acteurExterneAl'IUEM* : Identifiant numérique permettant de référencer un acteur externe à l'**IUEM**.
- *public string adresse* : Adresse de l'acteur.

Classe "acteurInterneAl'IUEM"

Description

Les acteurs internes à l'**IUEM** sont les acteurs qui font partis de l'**IUEM** de Plouzané : c.a.d. les membres des équipes de recherche, les doctorants, mais aussi les étudiants de Maîtrise, DEA, Master 1&2.

Classes parentes :

- *acteur*

Attributs :

- *public integer id_acteurInterneAl'IUEM* : Identifiant numérique permettant de référencer un acteur interne à l'**IUEM**.
- *public Set(0..1) integer[4] num_poste* : Numéro d'identification d'un poste.
- *public Set(0..1) string num_bureau* : Numéro d'identification d'un bureau.

Classe "production"

Description

La classe "production" représente un des types de documentation de la mémoire environnementale. En fait, elle contient toutes les productions (sous de multiples formes) obtenus grâce aux données environnementale, mais aussi les productions utilisés au cours d'un projet.

Contraintes : {xor}

Classes parentes :

- *documentation*

Types externes :

- *production*

Attributs :

- *public integer id_production* : Identifiant numérique permettant de référencer une production.

- *public Set(0..1) production production_source_propre* : La source d'une production est l'attribut qui permet de chaîner les productions entre elles notamment pour permettre de constituer la bibliographie associée à une production.
- *public string date_production* : Date de création de la production. (Cette date est renseignée par la personne qui crée la production : ceci pour permettre de dater la production au jour de sa création et non au jour de son insertion dans le système).
- *public string titre_production* : Titre/intitulé de la production.
- *public Set(0..1) string url_production* : Adresse pour créer le lien avec le fichier contenant la production.

Classe "acteurTransition"

Description

Pendant un état des changements sur un acteur peuvent intervenir, comme l'ajout de compétences par exemple. La classe acteurTransition va permettre de typer et d'archiver ces modifications grâce à des libellés de transition et une date.

Attributs :

- *public integer id_acteurTransition* : Identifiant numérique permettant de référencer une transition d'un acteur.
- *public string libelle_acteurTransition* : Libellé d'une transition d'un acteur.

Classe "projetTransition"

Description

Cette classe regroupe les différentes transitions que peut avoir un projet. C'est à dire les libellés des actions réalisables au cours d'un projet.

Attributs :

- *public integer id_projetTransition* : Identifiant numérique permettant de référencer une transition de projet.
- *public string libelle_projetTransition* : Libellé d'une transition de projet.

Classe "projetEtat"

Description

Cette classe de type énuméré regroupe le libellé des différents états que peut avoir un projet.

Stéréotype : "enum"

Types énumérés :

- *projetEtat (en période d'activité, suspendu, stoppé)*

Classe "equipeDeRattachement"

Description

La classe equipeDeRattachement constitue un catalogue d'équipe à laquelle chaque acteur est attaché. Cette équipe peut être une équipe de recherche, une équipe à l'intérieur d'une équipe de recherche, le pôle ou le département d'une entreprise, une formation universitaire, ... ; comme : UMR 6554 LETG, 9 DEA, ...

Attributs :

- *public integer id_equipeDeRattachement* : Identifiant numérique permettant de référencer une équipe de rattachement.
- *public string libelle_equipeDeRattachement* : Nom de l'équipe de rattachement.
- *public Set(0..1) string url_equipeDeRattachement* : Lien avec le site de l'équipe.

Classe "liaisonSrcExtProjet"

Description

Classe d'association qui permet de dater et de commenter la relation "projet-projet", autrement dit l'ajout d'une source externe (référence) à un projet.

Attributs :

- *public string date_liaisonProjet* : Date de liaison entre les projets.
- *public string commentaires_liaisonProjet* : Commentaires de la liaison, explications du choix de lier les projets.

Classe "refBiblio"***Description***

Cette classe rassemble les éléments (attributs) nécessaires pour décrire une référence bibliographique.

Stéréotype : "struct"

Types externes :

- *refBiblio*

Attributs :

- *public integer nbre_de_pages* : Nombre de pages.
- *public string abstract* : Résumé.
- *public Set(0..1) string notes* : Notes sur l'écrit pour montrer son intérêt ou petite description de celui-ci.
- *public integer volume* : Volume.

Classe "liaisonSrcExtProduction"***Description***

Classe d'association qui permet de dater et de commenter la relation "production-production", autrement dit l'ajout d'une source externe (référence) à une production.

Attributs :

- *public string date_liaisonProduction* : Date de liaison entre les productions.
- *public string commentaires_liaisonProduction* : Commentaires de la liaison, explications du choix de lier les productions.

Classe "historiqueEtatActeur"***Description***

Cette classe permet d'archiver les différents états qu'un acteur a lors de sa présence dans le système. C'est-à-dire qu'à chaque changement d'état d'un acteur, une instance d'historiqueEtatActeur va être créée. Cette instance conserve le nom du nouvel, la date à laquelle le changement a eu lieu et commentaire pour argumenter ce changement. Bien sûr une instance de cette classe est liée avec une unique instance d'acteur.

Attributs :

- *public integer id_historiqueEtatActeur* : Identifiant numérique permettant de référencer une instance d'historiqueEtatActeur.
- *public acteurEtat nom_etat_acteur* : Nom de l'état courant de l'acteur.
- *public string date_etat_acteur* : Date à laquelle l'acteur change d'état.
- *public string commentaire_etat_acteur* : Commentaires sur l'état de l'acteur.

Classe "liaisonEquipeDeRattachement"***Description***

La classe d'association liaisonEquipeDeRattachement effectue un lien entre l'historique des états d'un acteur et les équipes de rattachement en apportant des attributs qui qualifient le lien comme une date de liaison, une date de rupture entre les deux classes (+les commentaires associés), mais aussi un statut et une provenance. A noter que la provenance de l'acteur n'est pas forcément la même que celle attribuée à son équipe de rattachement d'où la relation entre "provenance" et "liaisonEquipeDeRattachement".

Attributs :

- *public string date_liaison_equipe_de_rattachement* : Date à laquelle la liaison entre "historiqueEtatActeur" et "equipeDeRattachement" a eu lieu.
- *public string commentaire_liaison_equipe_de_rattachement* : Commentaires à propos de la liaison entre "historiqueEtatActeur" et "equipeDeRattachement".
- *public string date_rupture_equipe_de_rattachement* : Date à laquelle la rupture entre "historiqueEtatActeur" et "equipeDeRattachement" a eu lieu.
- *public string commentaire_rupture_equipe_de_rattachement* : Commentaires à propos de la rupture entre "historiqueEtatActeur" et "equipeDeRattachement".

3.3. Analyse des besoins

L'analyse des besoins des utilisateurs est une étape capitale dans la conception d'un système comme le nôtre. Lors des interviews certaines questions étaient plus particulièrement centrées sur ce sujet. Le dépouillement de ces entretiens nous a permis d'établir « une hiérarchie » des types de documentation.

Une des difficultés, dans la schématisation des cas d'utilisations, est de trouver un juste équilibre pour faire apparaître la meilleure information à l'aide d'un nombre de cas raisonnable. C'est pourquoi cette analyse a été découpée en deux parties, la première pour les cas d'utilisations dits de "gestion" et la deuxième pour les cas dits de "consultation".

3.3.1. Schémas des cas d'utilisations (CU)

Le schéma des cas d'utilisations permet de formaliser d'un point de vue global les fonctionnalités souhaitées par les utilisateurs pour le système. Il permet d'indiquer les acteurs qui auront accès à celles-ci, ceci est symbolisé par un lien.

Dans notre cas, nous avons plusieurs niveaux d'acteurs pour lesquels il est possible d'établir une hiérarchie afin qu'ils héritent les uns des autres des droits d'utilisations (voir ci-contre).

Les cas d'utilisations sont généralement divisés en deux catégories : ceux dits de "gestion" et ceux dits de "consultation".

- Les cas d'utilisations dits de "gestion" regroupent eux-mêmes des "sous-cas" d'utilisations qui sont principalement la création, la modification et la suppression, comme par exemple la gestion d'un projet comprend à la fois la création, la modification et la suppression d'un projet.
- les cas d'utilisations dits de "consultation" sont, comme le nom l'indique, des fonctionnalités qui permettent la consultation, dans notre cas ce sera principalement de l'affichage dynamisé par des liens hypertextes.

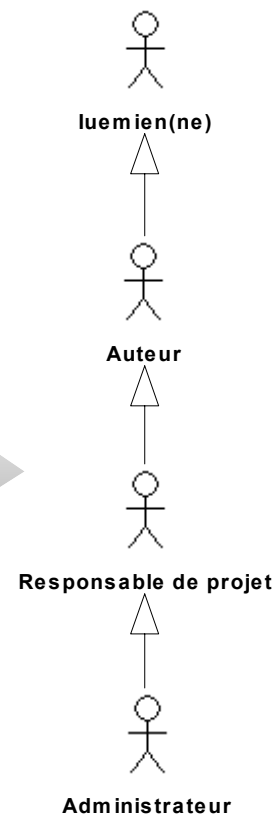


Figure 15 : Hiérarchie des acteurs.

Schéma pour les cas d'utilisations dits de « gestion » :

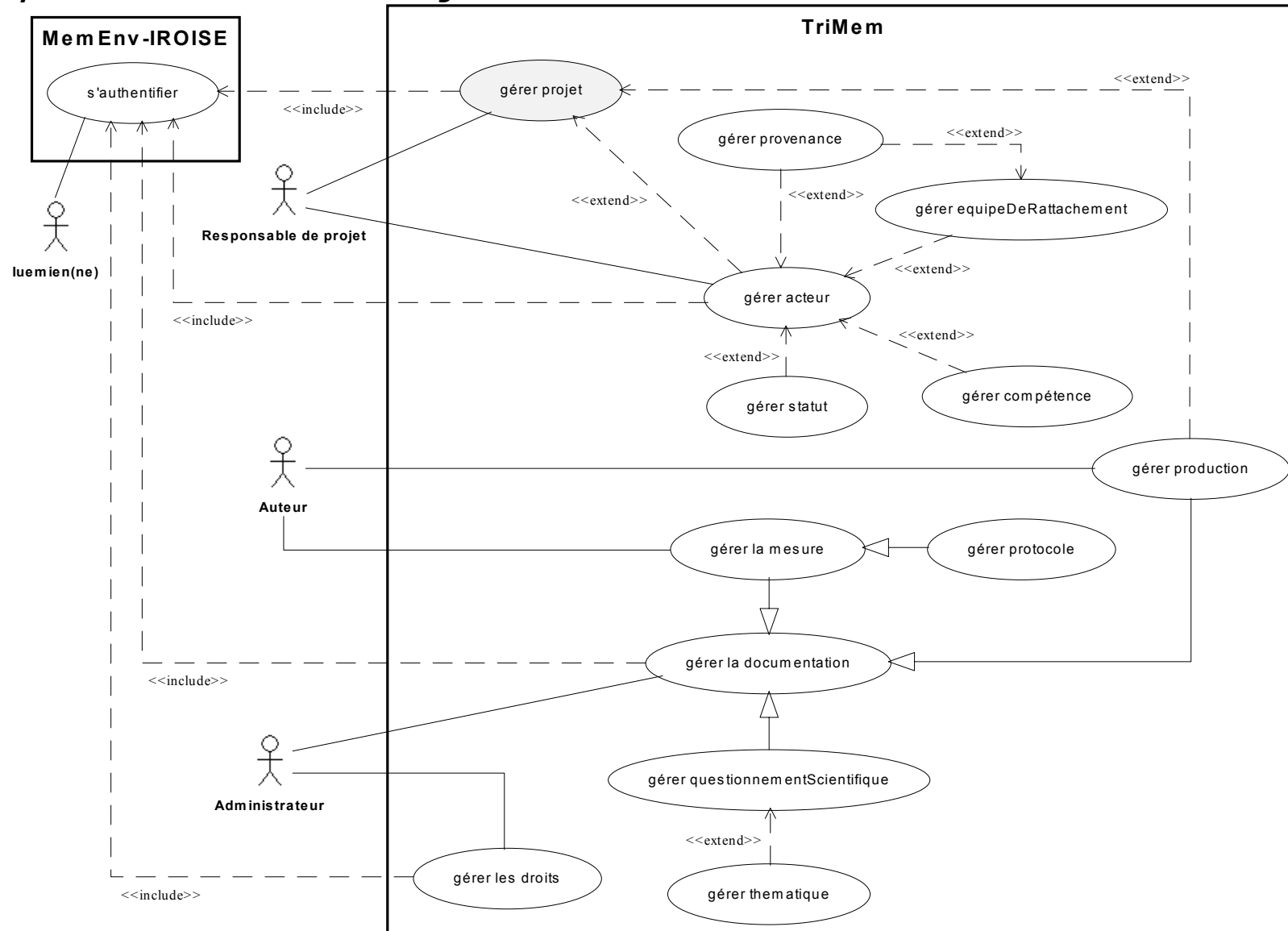


Figure 16 – Cas d'utilisations dits de « gestion ».

Schéma pour les cas d'utilisations dits de « consultation » :

Ce schéma représente les cas d'utilisations spécifiques à l'activité de consultation du système. Il fait intervenir les deux packages Denv pour la partie "donnée" avec la consultation du catalogue et des données environnementales. Le deuxième package est TriMem, pour la consultation de la documentation (productions), des projets et des acteurs. Les CU, contenus dans le cadre MemEnv-IROISE, sont valables pour l'ensemble des mémoires.

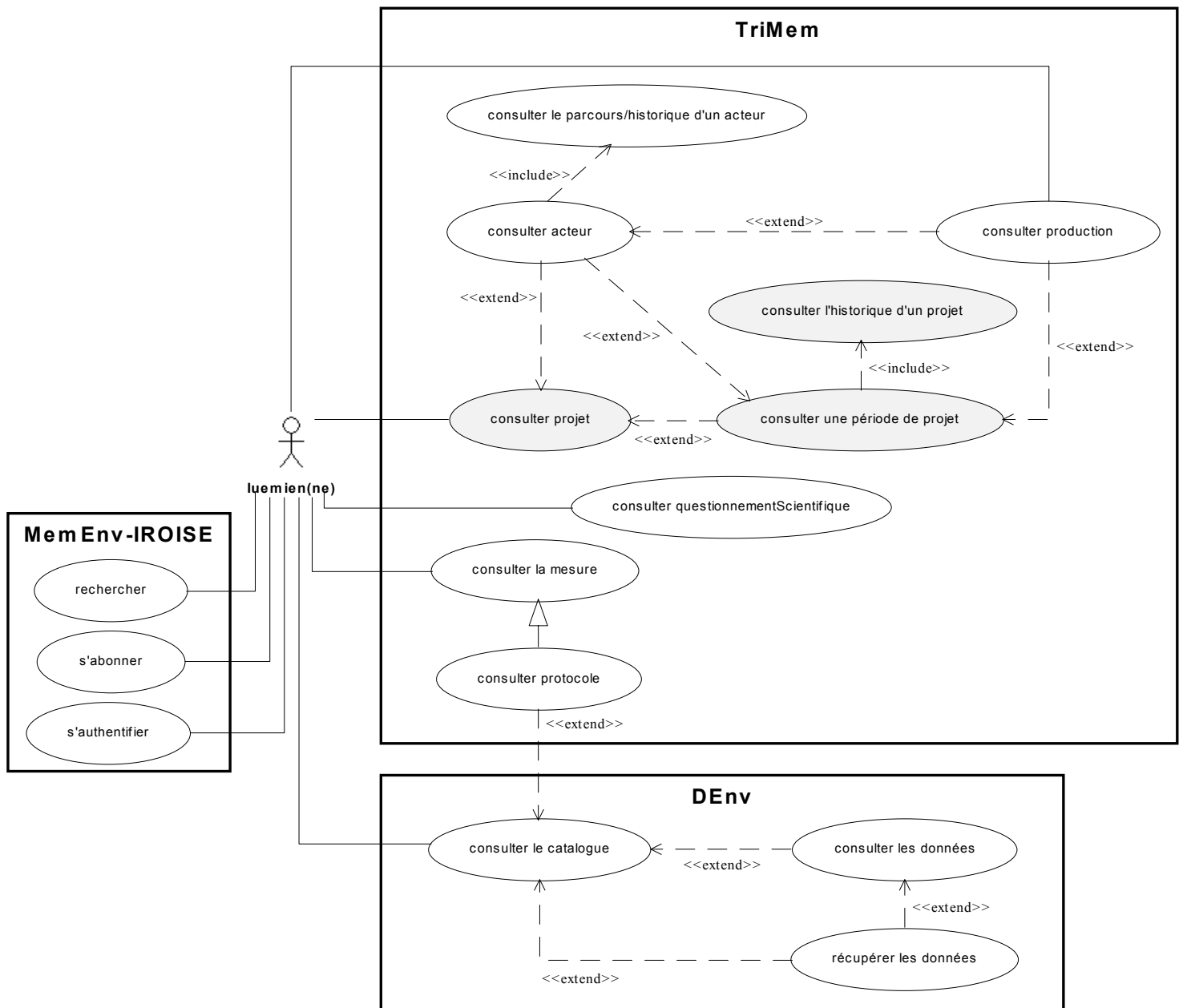


Figure 17 – Cas d'utilisations dits de « consultation ».

3.3.2. Description générale des C.U.

Le tableau qui suit, contient les descriptions des cas d'utilisations qui ont été vus aux *figures 16 et 17*. Chaque C.U. possède un acteur, une priorité de réalisation, un nom et une description. Comme précédemment les C.U. sont rangés selon le package auquel ils appartiennent.

	ACTEUR	PRIORITE	NOM CU	DESCRIPTION SYNTHETIQUE
PACKAGE MemEnv- IROISE	<i>luemien(ne)</i>	+	s'authentifier	L'authentification doit permettre à chaque utilisateur possédant un login et un mot de passe d'accéder aux fonctionnalités pour lesquelles il a obtenu les droits.
	<i>luemien(ne)</i>	+	rechercher	La fonctionnalité " <i>rechercher</i> " permet d'effectuer une recherche par mots-clés dans le système, tant pour les acteurs que pour les projets et les productions.
	<i>luemien(ne)</i>	-	s'abonner	Le but de " <i>s'abonner</i> " est d'offrir la possibilité aux utilisateurs de s'inscrire sur une liste de diffusion qui va les avertir par exemple de l'arrivée de nouvelles données. Il suffira de pré-sélectionner la/les donnée(s) voulue(s).
PACKAGE TriMem	<i>Administrateur</i>	+	gérer la documentation	La gestion de la documentation permet de choisir quel type de fonctionnalité l'on souhaite utiliser : c'est-à-dire gérer le questionnement Scientifique, la production ou bien encore la mesure.
	<i>Auteur</i>	++	gérer production	La gestion de la production permet de choisir sur quel type de production l'on souhaite travailler pour ensuite être dirigé vers un des cas d'utilisations qui héritent de " <i>gérer production</i> ".
	<i>Administrateur</i>	-	gérer questionnementScientifiqu e	La gestion d'un questionnement scientifique offre la possibilité de créer, de supprimer ou de modifier un questionnement en lui-même et chacun de ses attributs. Elle comprend également l'affectation de une ou plusieurs thématiques pour chaque questionnement.
	<i>Auteur</i>	-	gérer la mesure	La gestion de la mesure offre la possibilité de se diriger vers la gestion de protocole.



<i>Responsable de projet</i>	++	gérer projet	<p>La gestion d'un projet offre la possibilité de créer, de supprimer ou de modifier un projet en lui-même et chacun de ses attributs. Elle comprend également l'affectation :</p> <ul style="list-style-type: none"> - de participants internes et/ou externes (avec le choix des responsabilités), - de productions produites grâce aux travaux du projet, - de productions étant considérées comme des sources bibliographiques, - de sources de données environnementales, - d'un état et de transitions pour assurer la constitution d'un historique clair et détaillé. <p>Afin de visualiser le résultat de la saisie et de le valider, une fiche rassemblant tous les éléments est proposée à la fin.</p>
<i>Responsable de projet</i>	+	gérer acteur	<p>La gestion d'un acteur offre la possibilité de créer, de supprimer ou de modifier un acteur en lui-même et chacun de ses attributs. Elle comprend également l'affectation d'un statut, d'une provenance en fonction d'une équipe de rattachement, en sachant qu'un acteur peut avoir plusieurs équipes de rattachement. Ce cas d'utilisation permet également de créer de nouveaux états pour un acteur afin de constituer son historique.</p>
<i>Responsable de projet</i>	-	gérer equipeDeRattachement	<p>La gestion d'une équipe de rattachement offre la possibilité de créer, de supprimer ou de modifier une équipe en elle-même et chacun de ses attributs. Elle comprend également l'affectation d'un type d'équipe de rattachement et d'une provenance pour chaque équipe.</p>
<i>Responsable de projet</i>	-	gérer provenance	<p>La gestion d'une provenance offre la possibilité de créer, de supprimer ou de modifier une provenance en elle-même et chacun de ses attributs.</p>
<i>Responsable de projet</i>	-	gérer compétence	<p>La gestion d'une compétence offre la possibilité de créer, de supprimer ou de modifier une compétence en elle-même et chacun de ses attributs. Elle comprend également l'affectation d'un type de compétence pour chaque compétence.</p>
<i>Administrateur</i>	++	gérer les droits	<p>La gestion des droits consiste à affecter des droits sur les tables à tous les utilisateurs et à sécuriser l'accès à la/les base(s) de données.</p>
<i>Auteur</i>	-	gérer protocole	<p>La gestion d'un protocole comprend la création, la suppression et la modification d'un protocole en lui-même et chacun de ses attributs.</p>
<i>Administrateur</i>	-	gérer thématique	<p>La gestion d'une thématique offre la possibilité de créer, de supprimer ou de modifier une thématique en elle-même et chacun de ses attributs.</p>



	<i>luemien(ne)</i>	++	consulter production	La consultation des productions consiste à afficher les références d'une production, ainsi que sa bibliographie, le projet producteur et pourquoi pas tous les projets qui l'utilisent.
	<i>luemien(ne)</i>	+	consulter projet	Affiche la dernière période des projets de chaque projet en distinguant les projets qui sont en période d'activité, ceux qui sont suspendus et ceux qui sont stoppés.
	<i>luemien(ne) selon les autorisations</i>	++	consulter une période de projet	Affiche les caractéristiques (attributs) d'un projet, les participants au projet, les productions, les sources du projet (bibliographie), ...
	<i>luemien(ne) selon les autorisations</i>	++	consulter l'historique d'un projet	La consultation de l'historique d'un projet consiste à afficher ses différentes périodes (états) au cours du temps et les transitions qui sont associées.
	<i>luemien(ne)</i>	++	consulter acteur	La consultation des acteurs consiste à visualiser les caractéristiques (attributs) des acteurs, mais aussi les compétences, les statuts, les provenances en fonction des équipes de rattachement.
	<i>luemien(ne)</i>	++	consulter le parcours/historique d'un acteur	La consultation du parcours d'un acteur consiste à afficher ses différents états (enchaînements) au cours du temps et les transitions qui sont associées ; plus les productions et les projets auxquels l'acteur participe.
	<i>luemien(ne)</i>	-	consulter questionnaire Scientifique	Permet de consulter les questionnements scientifiques et les thématiques associées.
	<i>luemien(ne)</i>	-	consulter la mesure	Permet de visualiser l'ensemble des protocoles disponibles
		-	consulter protocole	Permet de consulter un protocole de mesure
PACKAGE D'Env	<i>luemien(ne)</i>	++	consulter le catalogue	La consultation du catalogue consiste à visualiser les données environnementales disponibles au sein du système. C'est à partir du catalogue que l'on peut être redirigé pour visualiser et/ou récupérer des données.
	<i>luemien(ne) selon les autorisations</i>	++	consulter les données	Ce cas d'utilisation permet de visualiser des données environnementales, le choix de la donnée à afficher doit être fait au préalable : c'est à dire à partir du catalogue. Ici s'effectue les choix de paramètres, périodes et positions géographiques afin de cibler les données au plus près.
	<i>luemien(ne) selon les autorisations</i>	+	récupérer les données	Ce cas d'utilisation permet de récupérer des données environnementales, le choix de la donnée à retirer doit être fait au préalable : c'est à dire à partir du catalogue. Ici s'effectue les choix de paramètres, périodes et positions géographiques afin de cibler les données au plus près.

3.3.3. Etude détaillée du C.U. « Gérer un projet »

Objectifs :

La gestion d'un projet offre la possibilité à un responsable de projet ou à l'administrateur de créer, de supprimer ou de modifier un projet (ou une période d'un projet) en lui-même et chacun de ses attributs. Elle comprend également l'affectation :

- de participants internes et/ou externes (avec le choix des responsabilités),
- de productions produites grâce aux travaux du projet,
- de productions étant considérées comme des sources bibliographiques,
- de sources de données environnementales,
- d'un état et de transitions pour assurer la constitution d'un historique clair et détaillé.

Afin de visualiser le résultat de la saisie et de le valider, une fiche rassemblant tous les éléments est proposée à la fin.

Description détaillée :

// L'accès à ce cas d'utilisations se fait à partir du cadre " menu " : le système affiche les différentes fonctionnalités (CU) dont " Projets ", à partir d'ici on choisit " Gérer les projets " qui se décline lui même en " Créer un projet ", " Créer une nouvelle période " et " Modifier une période ".

// L'Interaction Homme-Machine (IHM) de la fonctionnalité choisie se trouve dans le cadre " contenu ".

// L'utilisateur doit s'être authentifié auparavant.

// Objet utilisé : "l'explorer" de projet : il permet d'afficher et de sélectionner les périodes de projets. L'affichage et la navigation y sont facilités grâce à la forme en arbre.

Scénario nominal :

1) Le système affiche une liste de choix avec trois possibilités : créer un nouveau projet, créer une nouvelle période et modifier une période. L'utilisateur peut également faire une demande de suppression pour une période d'un projet ; la suppression, si elle est motivée, sera effectuée seulement par le WebMaster. L'utilisateur choisit la fonctionnalité désirée.

2) 1er cas : " créer un nouveau projet " a été choisi : une barre d'étape apparaît avec les différentes étapes à réaliser.

2ème cas : " créer une nouvelle période d'un projet " a été choisi : " l'explorer " de projet s'affiche et l'utilisateur doit sélectionner le projet auquel il souhaite ajouter une période. Après avoir validé cette sélection, une barre d'étape apparaît avec les différentes étapes à réaliser et les caractéristiques du projet source (nom, objectifs, enjeux, ...) sont affichées.

3ème cas : " modifier une période d'un projet " a été choisi : " l'explorer " de projet s'affiche et l'utilisateur doit sélectionner la période qu'il souhaite modifier. Après avoir validé cette sélection, une barre d'étape apparaît avec les différentes étapes à réaliser et tous les paramètres de la période sont disponibles.

4ème cas : " Faire une demande de suppression d'une période ou d'un projet " a été choisi : l'IHM de demande de suppression s'affiche dans une nouvelle fenêtre. Le demandeur doit fournir son prénom, nom, mail, les motivations de cette demande et il sélectionne la période à supprimer grâce à " l'explorer " de projet. Puis il valide l'envoi de la demande au WebMaster et il est redirigé vers le menu l'IHM " Gérer les projets ".

3) Pour les trois premiers cas du point précédent, une barre d'étape s'affiche, cette barre permet de naviguer entre les différentes étapes de construction d'un projet qui sont :

- Saisir les caractéristiques
- Affecter les participants
- Affecter les productions
- Affecter les données
- Affecter les compétences
- Affecter les sources bibliographiques
- Affecter les documents de suivi de projet
- Affecter les sources projets externes

Détail des étapes :

Saisir les caractéristiques : il s'agit de saisir le nom, objectifs, enjeux, commentaires, ... sur le projet.

Affecter les participants : l'utilisateur doit sélectionner les participants au projet et définir le(s) responsable(s) du projet.

Affecter les productions : ceci permet de répertorier les productions issues du projet, la production doit être préalablement créée.

Affecter les données : il s'agit de sélectionner les données environnementales utilisées lors du projet.

Affecter les compétences : il s'agit de sélectionner les compétences que l'on retrouve au sein des participants projet.

Affecter les sources bibliographiques : afin d'établir la bibliographie du projet, l'utilisateur va sélectionner les productions utilisées comme sources.

Affecter les documents de suivi de projet : au cours du projet, de la documentation de suivi de projet autre qu'une publication ou un rapport est produite et donc l'utilisateur peut référencer ces documents.

Affecter les sources projets externes : un projet peut être issu d'un autre projet donc l'utilisateur peut attacher au projet d'autres projets.

Chaque étape est validée soit en cliquant sur " VALIDER ", soit en cliquant sur un composant de la barre d'étape.

4) La dernière étape : " Visualisation finale " permet de visualiser la forme du projet et l'action sur le bouton " VALIDER et ENREGISTRER LE PROJET " permet de valider la sauvegarde du document. Après avoir validé la sauvegarde, le cadre " menu " est mis à jour pour qu'il tienne compte de la modification effectuée, puis le système indique que la création ou la modification a bien eu lieu et propose de revenir au menu de l'IHM " Gérer les projets ".

Les cas d'erreurs :

Un message d'erreur s'affiche si un des champs obligatoires n'est pas renseigné.

Un message d'erreur s'affiche si un des champs renseignés est mal renseigné :

- mauvais type de paramètres,
- longueur excessive,
- incohérence des dates,
- ...

Les annulations :

A tout moment la saisie d'un projet peut être arrêtée et un message de confirmation de l'annulation apparaît, si l'utilisateur confirme, alors toute la saisie est perdue, c'est-à-dire qu'aucune requête d'insertion et de mise à jour ne seront faites dans la base de données. L'utilisateur est redirigé vers l'IHM " Gérer les projets ".

// L'utilisateur peut créer, modifier ou supprimer de nouveaux projets (périodes de), mais aussi retourner dans le menu pour consulter ou gérer d'autres composants du système.

Diagrammes de séquence :

Le premier diagramme de séquence (figure 16) permet simplement de montrer que l'utilisateur entre dans le cas d'utilisation, ensuite il choisit l'utilisation qu'il désire exécuter.

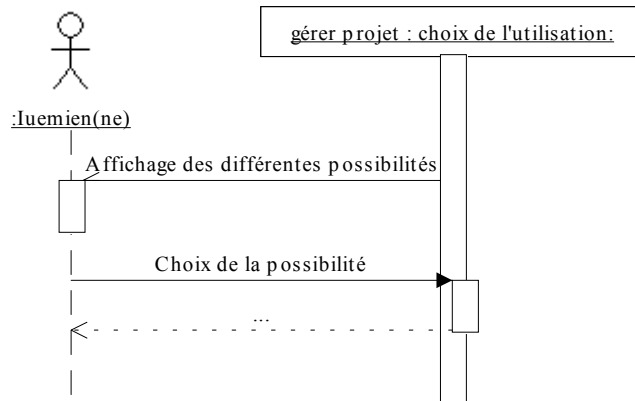


Figure 18 : Diagramme de séquence > "Gérer un projet : choix de l'utilisation".

Les quatre diagrammes de séquence qui suivent traitent des quatre scénarios nominaux possibles pour les sous-cas :

1. Création d'un projet
2. Création d'une nouvelle période de projet
3. Modification d'une période de projet
4. Suppression d'un projet

Chaque diagramme présente la séquencement des actions réalisées lors de l'exécution du CU et les échanges qui s'effectuent avec les différentes bases de données.

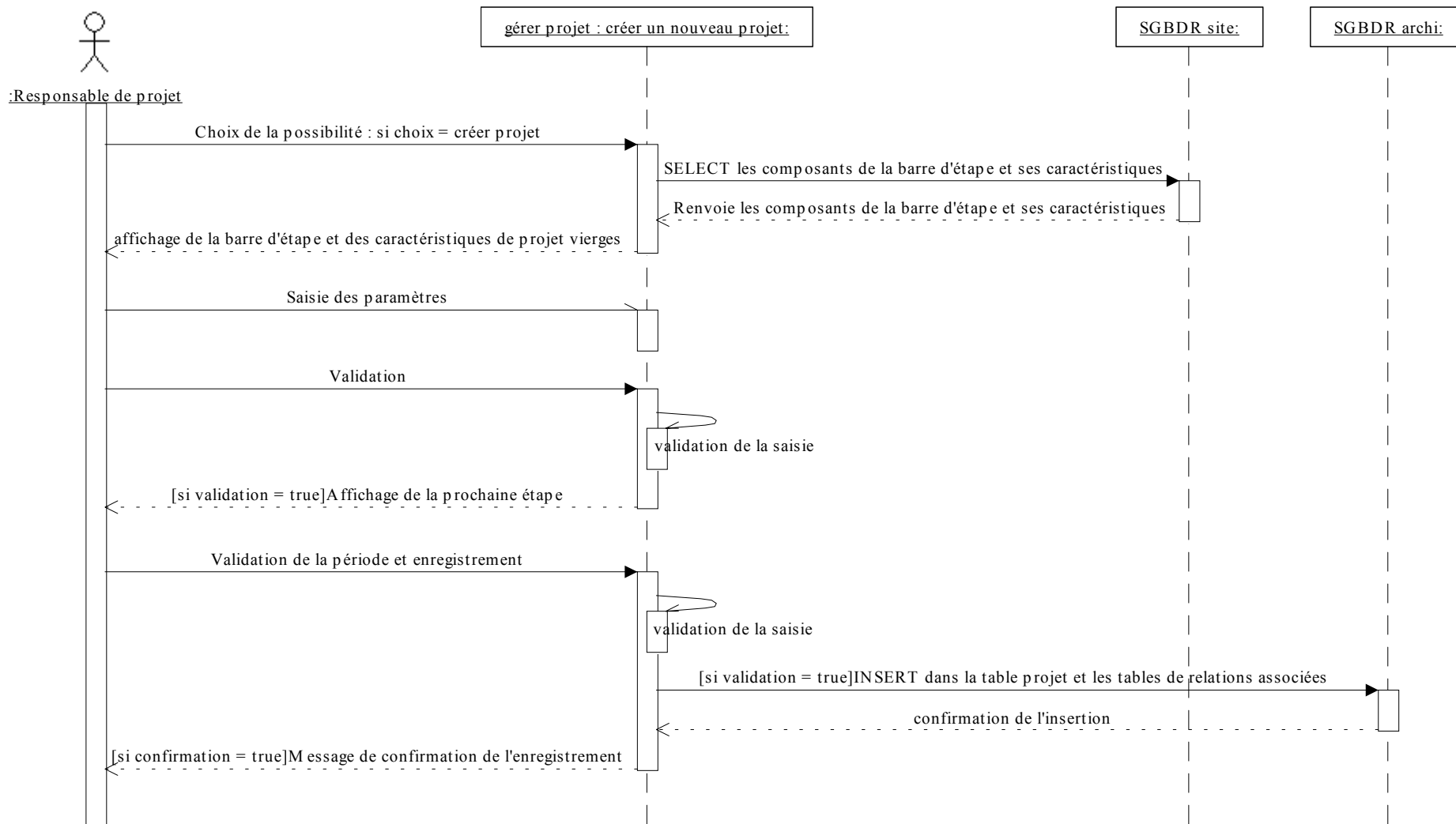


Figure 19 : Diagramme de séquence > "Gérer un projet : créer un nouveau projet".

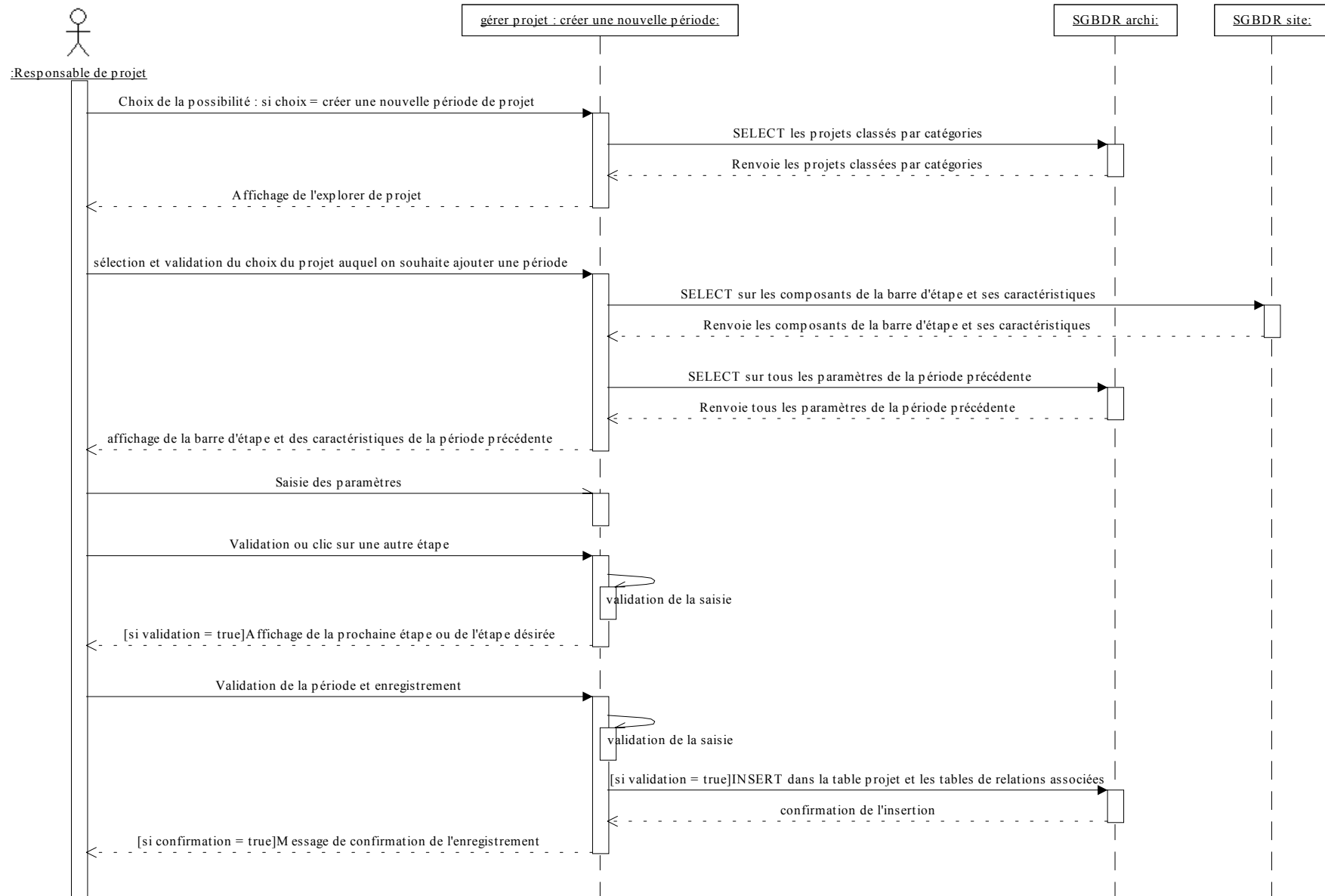


Figure 20 : Diagramme de séquence > "Gérer un projet : créer une nouvelle période".

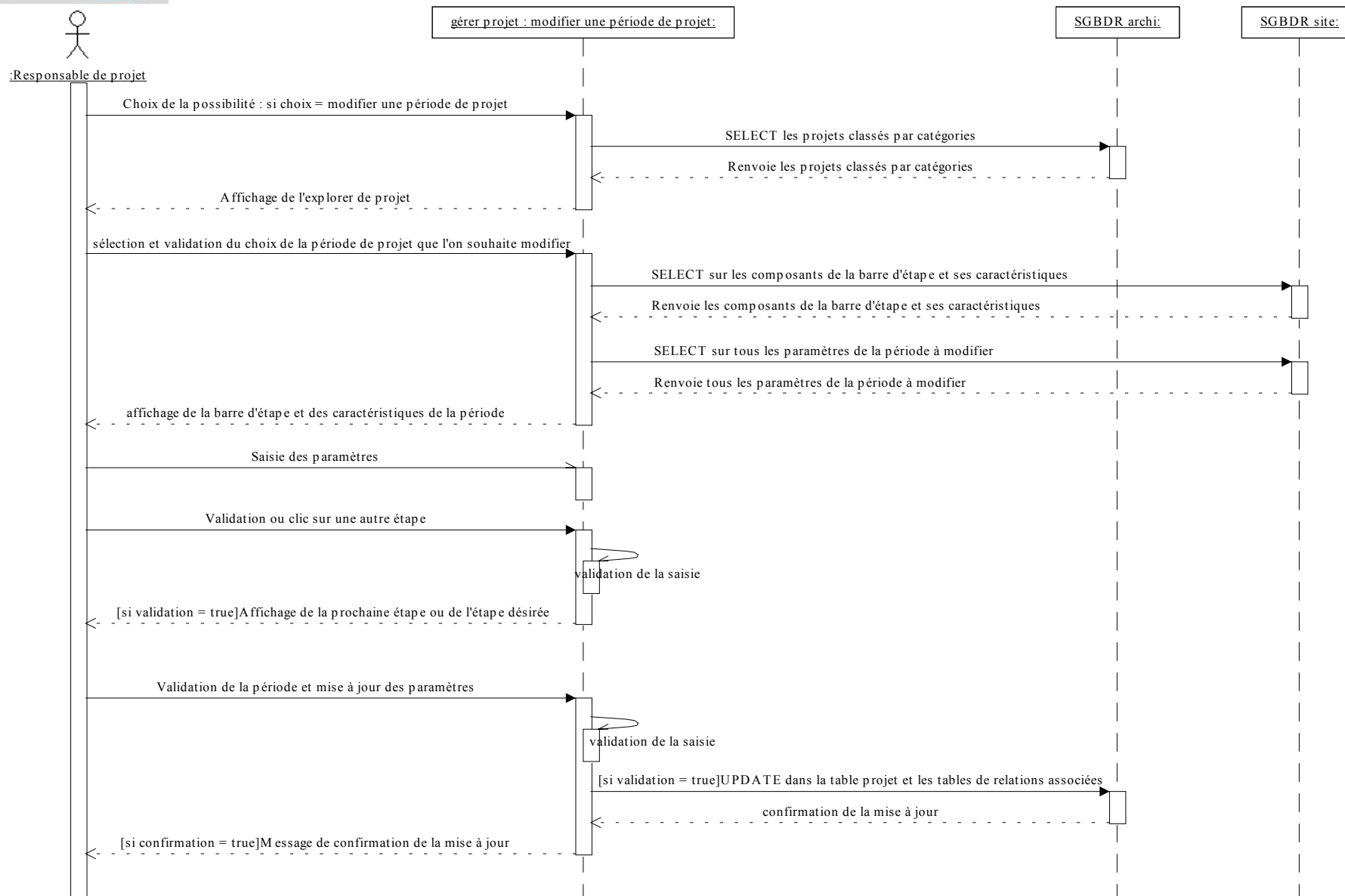


Figure 21 : Diagramme de séquence > "Gérer un projet : modifier une période".

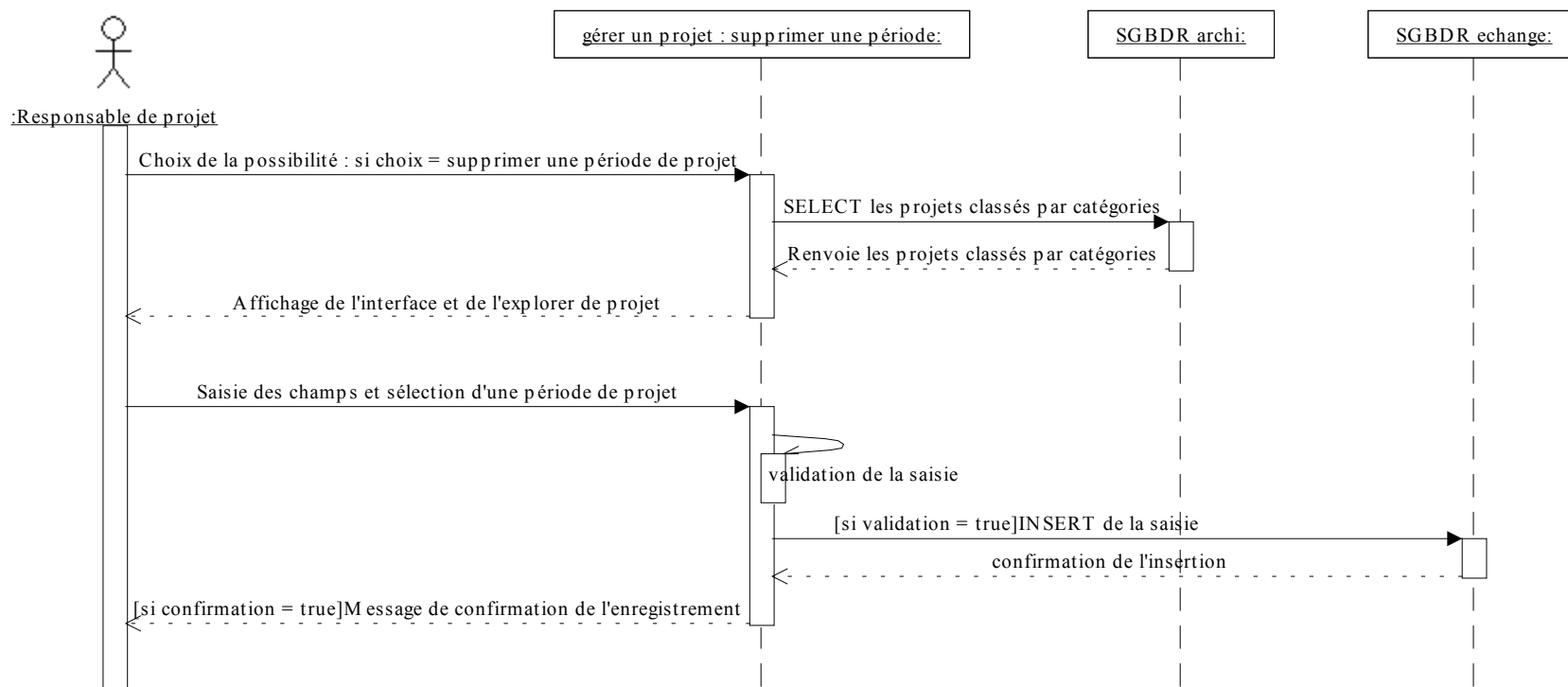


Figure 22 : Diagramme de séquence > "Gérer un projet : supprimer une période".

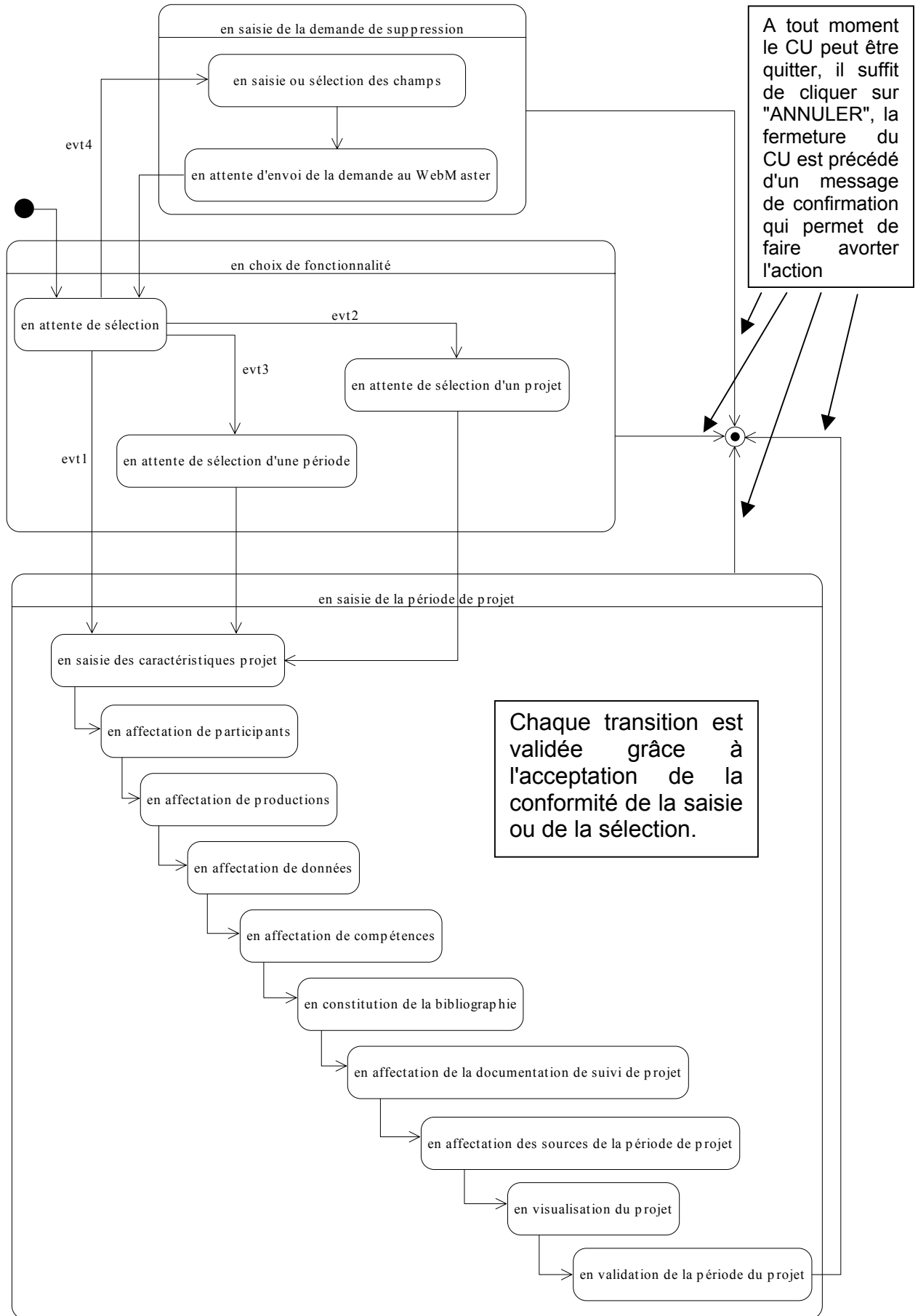


Figure 23 : Diagramme d'Etat-Transition de "Gérer un projet".

Description de l'IHM "Gérer les projets" :

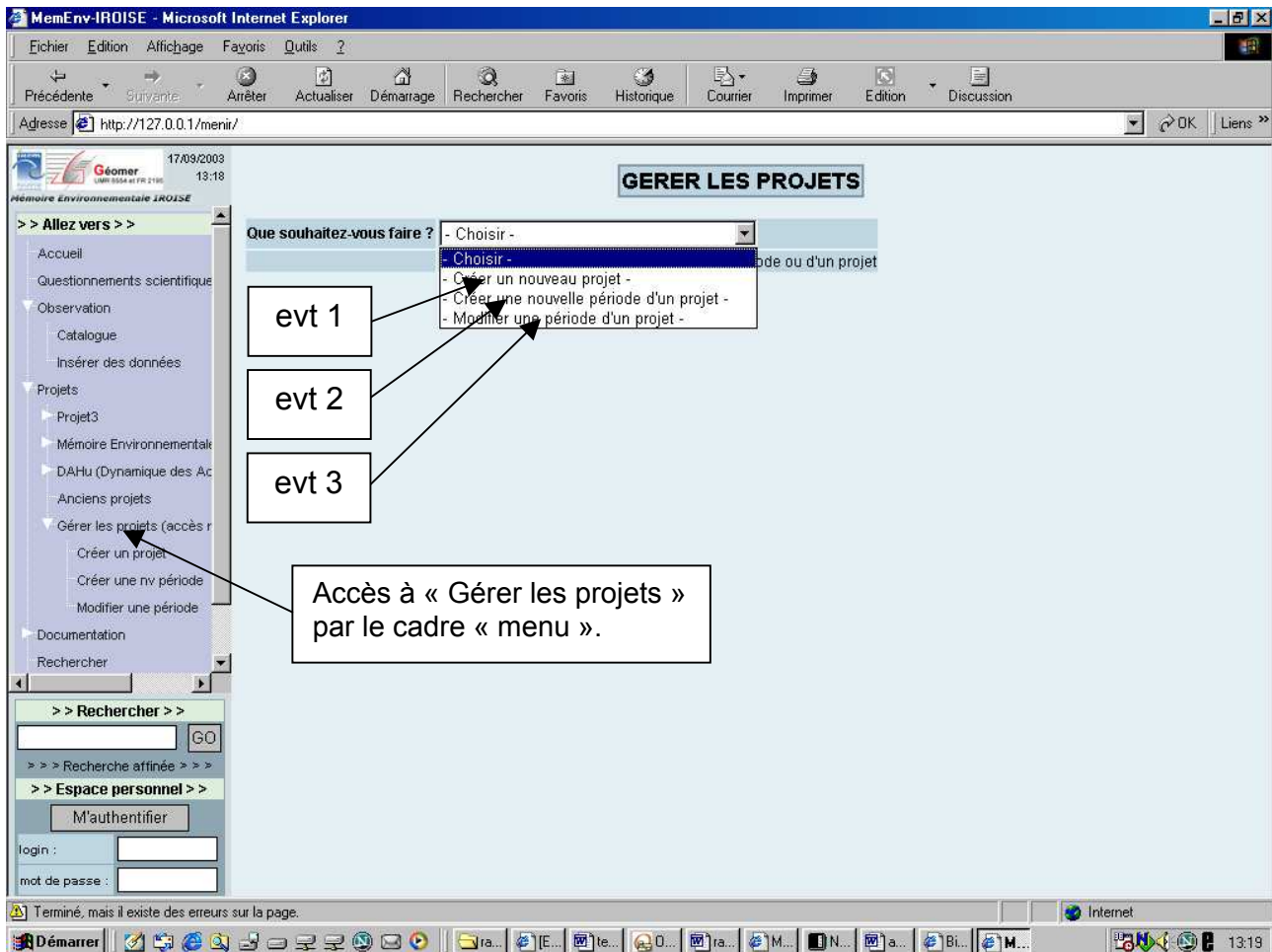


Figure 24 : Copie d'écran de l'IHM Gérer les projets pour les événements de 1 à 3.

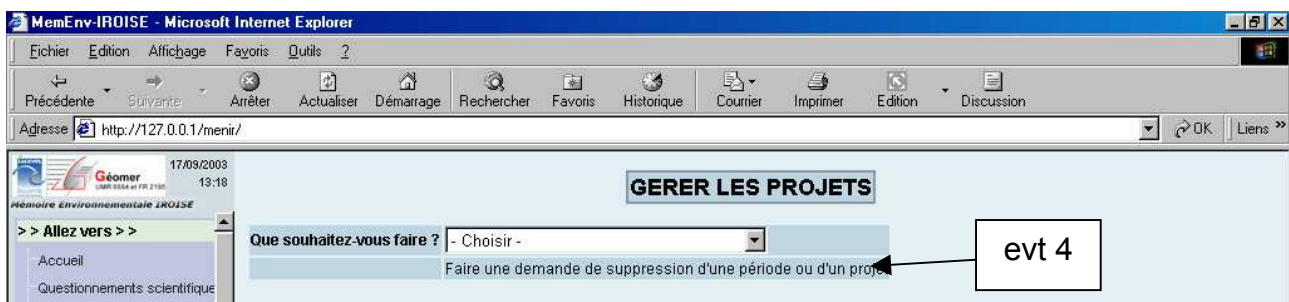


Figure 25 : Copie d'écran de l'IHM Gérer les projets pour l'évènement 4.

evt1 : permet de lancer le sous-cas "créer un nouveau projet".

evt2 : permet de lancer le sous-cas "créer une nouvelle période de projet".

evt3 : permet de lancer le sous-cas "modifier une période de projet".

evt4 : permet d'ouvrir le sous-cas "supprimer une période de projet" dans une nouvelle fenêtre.

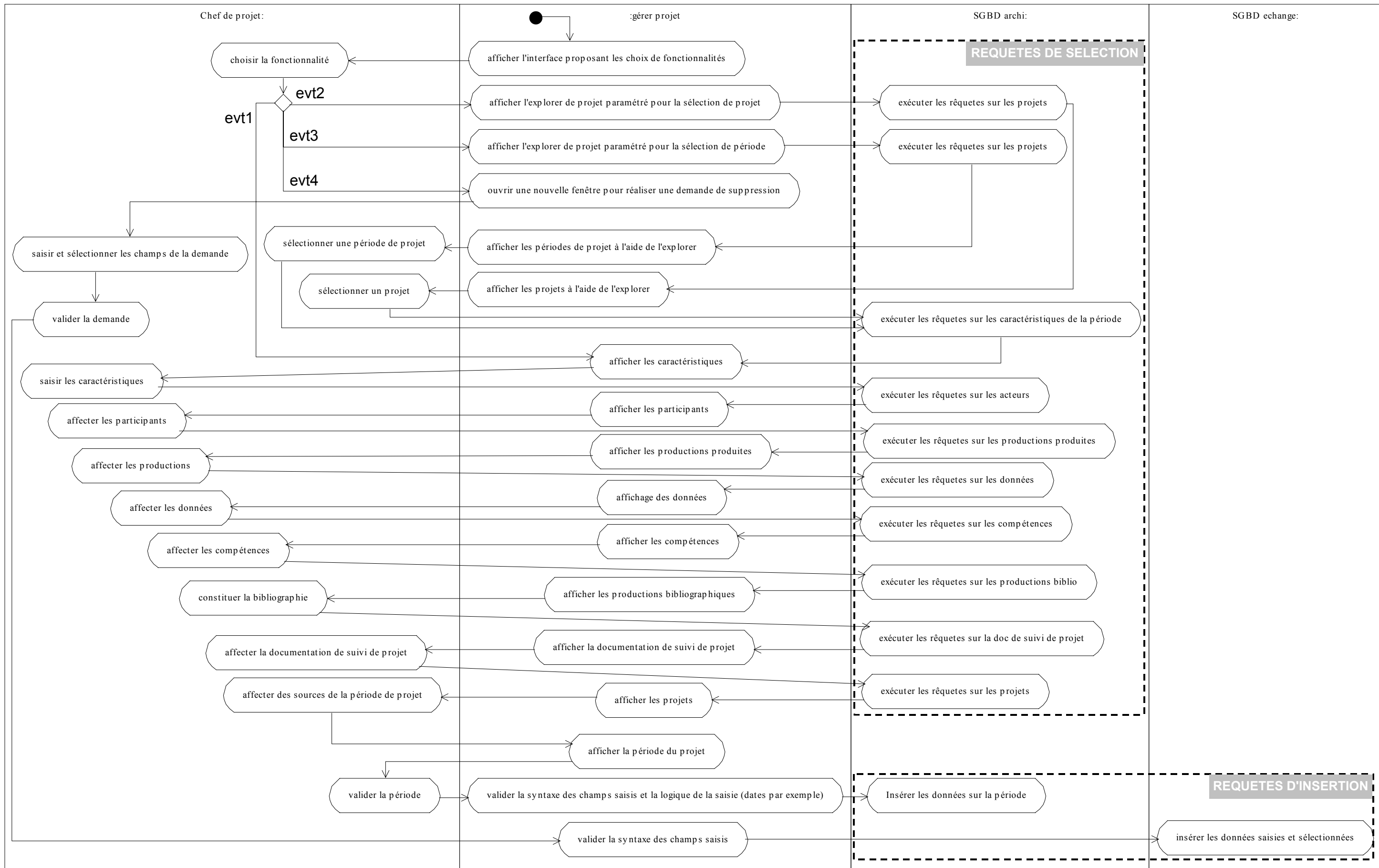
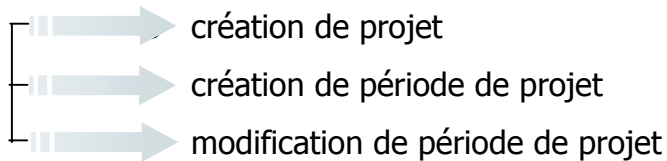


Figure 26 : Diagramme d'activité de "Gérer un projet".

Le diagramme d'activité du cas "Gérer un projet" (figure 26) montre toutes les actions qui vont être effectuées lors de l'exécution du C.U.. Ces actions sont ordonnées, le sens de lecture est donné par les flèches. Elles sont aussi réparties dans des couloirs. Un couloir correspond à un *acteur du système*, à un *cas d'utilisation* ou à un *composant* comme les bases de données "archi" et "échange". L'exécution du C.U. est une succession d'actions qui sont à la fois du ressort de l'homme pour la saisie des champs et de celui du code.

REQUETES SQL du CU "Gérer un projet"



Les requêtes qui suivent permettent d'insérer des projets, de les modifier et de les supprimer. Ces requêtes ont été générées pour toutes les tables qui sont en relation avec "projet" et qui participent au C.U. "Gérer un projet".

(Dans la base de données "archi")

- **Table PROJET**

```
INSERT INTO PROJET (  
ID_PROJET,  
NOM_PROJET,  
CONTEXTE_PROJET,  
ENJEUX_PROJET,  
OBJECTIFS_PROJET,  
HYPOTHESES_PROJET,  
COMMENTAIRES_PROJET,  
REF_PROJET,  
PROJET_ETAT_ID1,  
DATE_ETAT_PROJET,  
COMMENTAIRE_CHANGE_ETAT_PROJET,  
PROJET_SOURCE_PROPRES,  
APERCU_ETAT_PROJET,  
PERSPECTIVES_PROJET  
)  
VALUES (  
< ID_PROJET > ,  
"< NOM_PROJET >" ,  
"< CONTEXTE_PROJET >" ,  
"< ENJEUX_PROJET >" ,  
"< OBJECTIFS_PROJET >" ,  
"< HYPOTHESES_PROJET >" ,  
"< COMMENTAIRES_PROJET >" ,  
"< REF_PROJET >" ,  
< PROJET_ETAT_ID1 > ,  
"< DATE_ETAT_PROJET >" ,  
"< COMMENTAIRE_CHANGE_ETAT_PROJET >" ,  
< PROJET_SOURCE_PROPRES > ,
```

```
"< APERCU_ETAT_PROJET >" ,  
"< PERSPECTIVES_PROJET >"  
);
```

```
UPDATE PROJET SET ID_PROJET = < ID_PROJET > WHERE ID_PROJET = < NUMERO > ;  
UPDATE PROJET SET NOM_PROJET = < NOM_PROJET > WHERE ID_PROJET = < NUMERO > ;  
UPDATE PROJET SET CONTEXTE_PROJET = < CONTEXTE_PROJET > WHERE ID_PROJET = <  
NUMERO > ;  
UPDATE PROJET SET ENJEUX_PROJET = < ENJEUX_PROJET > WHERE ID_PROJET = < NUMERO >  
;  
UPDATE PROJET SET OBJECTIFS_PROJET = < OBJECTIFS_PROJET > WHERE ID_PROJET = <  
NUMERO > ;  
UPDATE PROJET SET HYPOTHESES_PROJET = < HYPOTHESES_PROJET > WHERE ID_PROJET = <  
NUMERO > ;  
UPDATE PROJET SET COMMENTAIRES_PROJET = < COMMENTAIRES_PROJET > WHERE  
ID_PROJET = < NUMERO > ;  
UPDATE PROJET SET REF_PROJET = < REF_PROJET > WHERE ID_PROJET = < NUMERO > ;  
UPDATE PROJET SET PROJET_ETAT_ID1 = < PROJET_ETAT_ID1 > WHERE ID_PROJET = <  
NUMERO > ;  
UPDATE PROJET SET DATE_ETAT_PROJET = < DATE_ETAT_PROJET > WHERE ID_PROJET = <  
NUMERO > ;  
UPDATE PROJET SET COMMENTAIRE_CHANGE_ETAT_PROJET = <  
COMMENTAIRE_CHANGE_ETAT_PROJET > WHERE ID_PROJET = < NUMERO > ;  
UPDATE PROJET SET PROJET_SOURCE_PROPRE = < PROJET_SOURCE_PROPRE > WHERE  
ID_PROJET = < NUMERO > ;  
UPDATE PROJET SET APERCU_ETAT_PROJET = < APERCU_ETAT_PROJET > WHERE ID_PROJET =  
< NUMERO > ;  
UPDATE PROJET SET PERSPECTIVES_PROJET = < PERSPECTIVES_PROJET > WHERE ID_PROJET  
= < NUMERO > ;
```

```
DELETE FROM PROJET  
WHERE ID_PROJET = < NUMERO > ;
```

- *Table R_PARTICIPANT_INTERNE*

```
INSERT INTO R_PARTICIPANT_INTERNE (  
ACTEUR_INTERNE_IUEM_ID1,  
PROJET_ID5,  
DATE_AR_PARTICIPANT_INTERNE,  
DATE_DEP_PARTICIPANT_INTERNE  
)  
VALUES (  
< ACTEUR_INTERNE_IUEM_ID1 > ,  
< PROJET_ID5 > ,  
"< DATE_AR_PARTICIPANT_INTERNE >" ,  
"< DATE_DEP_PARTICIPANT_INTERNE >"  
);
```

```
UPDATE R_PARTICIPANT_INTERNE SET ACTEUR_INTERNE_IUEM_ID1 = <  
ACTEUR_INTERNE_IUEM_ID1 > WHERE ACTEUR_INTERNE_IUEM_ID1 = < NUMERO > ;  
UPDATE R_PARTICIPANT_INTERNE SET PROJET_ID5 = < PROJET_ID5 > WHERE  
ACTEUR_INTERNE_IUEM_ID1 = < NUMERO > ;  
UPDATE R_PARTICIPANT_INTERNE SET DATE_AR_PARTICIPANT_INTERNE = <  
DATE_AR_PARTICIPANT_INTERNE > WHERE ACTEUR_INTERNE_IUEM_ID1 = < NUMERO > ;  
UPDATE R_PARTICIPANT_INTERNE SET DATE_DEP_PARTICIPANT_INTERNE = <  
DATE_DEP_PARTICIPANT_INTERNE > WHERE ACTEUR_INTERNE_IUEM_ID1 = < NUMERO > ;
```

```
DELETE FROM R_PARTICIPANT_INTERNE  
WHERE ACTEUR_INTERNE_IUEM_ID1 = < NUMERO > ;
```

- **Table R_PARTICIPANT_EXTERNE**

```
UPDATE R_PARTICIPANT_EXTERNE SET ACTEUR_EXTERNE_IUEM_ID1 = <
ACTEUR_EXTERNE_IUEM_ID1 > WHERE ACTEUR_EXTERNE_IUEM_ID1 = < NUMERO > ;
UPDATE R_PARTICIPANT_EXTERNE SET PROJET_ID7 = < PROJET_ID7 > WHERE
ACTEUR_EXTERNE_IUEM_ID1 = < NUMERO > ;
UPDATE R_PARTICIPANT_EXTERNE SET DATE_AR_PARTICIPANT_EXTERNE = <
DATE_AR_PARTICIPANT_EXTERNE > WHERE ACTEUR_EXTERNE_IUEM_ID1 = < NUMERO > ;
UPDATE R_PARTICIPANT_EXTERNE SET DATE_DEP_PARTICIPANT_EXTERNE = <
DATE_DEP_PARTICIPANT_EXTERNE > WHERE ACTEUR_EXTERNE_IUEM_ID1 = < NUMERO > ;
```

```
INSERT INTO R_PARTICIPANT_EXTERNE (
ACTEUR_EXTERNE_IUEM_ID1,
PROJET_ID7,
DATE_AR_PARTICIPANT_EXTERNE,
DATE_DEP_PARTICIPANT_EXTERNE
)
VALUES (
< ACTEUR_EXTERNE_IUEM_ID1 > ,
< PROJET_ID7 > ,
"< DATE_AR_PARTICIPANT_EXTERNE >" ,
"< DATE_DEP_PARTICIPANT_EXTERNE >"
);
```

```
DELETE FROM R_PARTICIPANT_EXTERNE
WHERE ACTEUR_EXTERNE_IUEM_ID1 = < NUMERO > ;
```

- **Table R_RESPONSABLE_INTERNE**

```
INSERT INTO R_RESPONSABLE_INTERNE (
ACTEUR_INTERNE_IUEM_ID2,
PROJET_ID4,
DATE_AR_RESPONSABLE_INTERNE,
DATE_DEP_RESPONSABLE_INTERNE
)
VALUES (
< ACTEUR_INTERNE_IUEM_ID2 > ,
< PROJET_ID4 > ,
"< DATE_AR_RESPONSABLE_INTERNE >" ,
"< DATE_DEP_RESPONSABLE_INTERNE >"
);
```

```
UPDATE R_RESPONSABLE_INTERNE SET ACTEUR_INTERNE_IUEM_ID2 = <
ACTEUR_INTERNE_IUEM_ID2 > WHERE ACTEUR_INTERNE_IUEM_ID2 = < NUMERO > ;
UPDATE R_RESPONSABLE_INTERNE SET PROJET_ID4 = < PROJET_ID4 > WHERE
ACTEUR_INTERNE_IUEM_ID2 = < NUMERO > ;
UPDATE R_RESPONSABLE_INTERNE SET DATE_AR_RESPONSABLE_INTERNE = <
DATE_AR_RESPONSABLE_INTERNE > WHERE ACTEUR_INTERNE_IUEM_ID2 = < NUMERO > ;
UPDATE R_RESPONSABLE_INTERNE SET DATE_DEP_RESPONSABLE_INTERNE = <
DATE_DEP_RESPONSABLE_INTERNE > WHERE ACTEUR_INTERNE_IUEM_ID2 = < NUMERO > ;
```

```
DELETE FROM R_RESPONSABLE_INTERNE
WHERE ACTEUR_INTERNE_IUEM_ID2 = < NUMERO > ;
```

- **Table R_CORESPONSABLE_EXTERNE**

```
INSERT INTO R_CORESPONSABLE_EXTERNE (
ACTEUR_EXTERNE_IUEM_ID2,
PROJET_ID6,
DATE_AR_CORESPONSABLE_EXTERNE,
```



```
DATE_DEP_CORESPONSABLE_EXTERNE  
)
```

```
VALUES (  
< ACTEUR_EXTERNE_IUEM_ID2 > ,  
< PROJET_ID6 > ,  
"< DATE_AR_CORESPONSABLE_EXTERNE >" ,  
"< DATE_DEP_CORESPONSABLE_EXTERNE >"  
);
```

```
UPDATE R_CORESPONSABLE_EXTERNE SET ACTEUR_EXTERNE_IUEM_ID2 = <  
ACTEUR_EXTERNE_IUEM_ID2 > WHERE ACTEUR_EXTERNE_IUEM_ID2 = < NUMERO > ;  
UPDATE R_CORESPONSABLE_EXTERNE SET PROJET_ID6 = < PROJET_ID6 > WHERE  
ACTEUR_EXTERNE_IUEM_ID2 = < NUMERO > ;  
UPDATE R_CORESPONSABLE_EXTERNE SET DATE_AR_CORESPONSABLE_EXTERNE = <  
DATE_AR_CORESPONSABLE_EXTERNE > WHERE ACTEUR_EXTERNE_IUEM_ID2 = < NUMERO > ;  
UPDATE R_CORESPONSABLE_EXTERNE SET DATE_DEP_CORESPONSABLE_EXTERNE = <  
DATE_DEP_CORESPONSABLE_EXTERNE > WHERE ACTEUR_EXTERNE_IUEM_ID2 = < NUMERO > ;
```

```
DELETE FROM R_CORESPONSABLE_EXTERNE  
WHERE ACTEUR_EXTERNE_IUEM_ID2 = < NUMERO > ;
```

- *Table R_LIAISON_COMPETENCE_PROJET*

```
INSERT INTO R_LIAISON_COMPETENCE_PROJET (  
PROJET_ID36,  
COMPETENCE_ID2,  
DATE_LIAISON_COMPETENCE36,  
COMMENTAIRE_LIAISON_COMPETENCE36  
)  
VALUES (  
< PROJET_ID36 > ,  
< COMPETENCE_ID2 > ,  
"< DATE_LIAISON_COMPETENCE36 >" ,  
"< COMMENTAIRE_LIAISON_COMPETENCE36 >"  
);
```

```
UPDATE R_LIAISON_COMPETENCE_PROJET SET PROJET_ID36 = < PROJET_ID36 > WHERE  
PROJET_ID36 = < NUMERO > ;  
UPDATE R_LIAISON_COMPETENCE_PROJET SET COMPETENCE_ID2 = < COMPETENCE_ID2 >  
WHERE PROJET_ID36 = < NUMERO > ;  
UPDATE R_LIAISON_COMPETENCE_PROJET SET DATE_LIAISON_COMPETENCE36 = <  
DATE_LIAISON_COMPETENCE36 > WHERE PROJET_ID36 = < NUMERO > ;  
UPDATE R_LIAISON_COMPETENCE_PROJET SET COMMENTAIRE_LIAISON_COMPETENCE36 = <  
COMMENTAIRE_LIAISON_COMPETENCE36 > WHERE PROJET_ID36 = < NUMERO > ;
```

```
DELETE FROM R_LIAISON_COMPETENCE_PROJET  
WHERE PROJET_ID36 = < NUMERO > ;
```

- *Table R_LIAISON_PROJET_TRANSITION_PROJET*

```
INSERT INTO R_LIAISON_PROJET_TRANSITION_PROJET (  
PROJET_ID3,  
PROJET_TRANSITION_ID1,  
DATE_TRANSITION_PROJET,  
COMMENTAIRE_TRANSITION_PROJET  
)  
VALUES (  
< PROJET_ID3 > ,  
< PROJET_TRANSITION_ID1 > ,
```

```
"< DATE_TRANSITION_PROJET >" ,  
"< COMMENTAIRE_TRANSITION_PROJET >"  
);
```

```
UPDATE R_LIAISON_PROJET_TRANSITION_PROJET SET PROJET_ID3 = < PROJET_ID3 > WHERE  
PROJET_ID3 = < NUMERO > ;  
UPDATE R_LIAISON_PROJET_TRANSITION_PROJET SET PROJET_TRANSITION_ID1 = <  
PROJET_TRANSITION_ID1 > WHERE PROJET_ID3 = < NUMERO > ;  
UPDATE R_LIAISON_PROJET_TRANSITION_PROJET SET DATE_TRANSITION_PROJET = <  
DATE_TRANSITION_PROJET > WHERE PROJET_ID3 = < NUMERO > ;  
UPDATE R_LIAISON_PROJET_TRANSITION_PROJET SET COMMENTAIRE_TRANSITION_PROJET =  
< COMMENTAIRE_TRANSITION_PROJET > WHERE PROJET_ID3 = < NUMERO > ;
```

```
DELETE FROM R_LIAISON_PROJET_TRANSITION_PROJET  
WHERE PROJET_ID3 = < NUMERO > ;
```

- **Table R_LIAISON_SERIE_DE_DONNEES_PROJET**

```
INSERT INTO R_LIAISON_SERIE_DE_DONNEES_PROJET (  
PROJET_ID80,  
SERIE_DE_DONNEES_ID80,  
DATE_LIAISON_SERIE_DE_DONNEES80,  
COMMENTAIRE_LIAISON_SERIE_DE_DONNEES80  
)  
VALUES (  
< PROJET_ID80 > ,  
< SERIE_DE_DONNEES_ID80 > ,  
"< DATE_LIAISON_SERIE_DE_DONNEES80 >" ,  
"< COMMENTAIRE_LIAISON_SERIE_DE_DONNEES80 >"  
);
```

```
UPDATE R_LIAISON_SERIE_DE_DONNEES_PROJET SET PROJET_ID80 = < PROJET_ID80 > WHERE  
PROJET_ID80 = < NUMERO > ;  
UPDATE R_LIAISON_SERIE_DE_DONNEES_PROJET SET SERIE_DE_DONNEES_ID80 = <  
SERIE_DE_DONNEES_ID80 > WHERE PROJET_ID80 = < NUMERO > ;  
UPDATE R_LIAISON_SERIE_DE_DONNEES_PROJET SET DATE_LIAISON_SERIE_DE_DONNEES80 =  
< DATE_LIAISON_SERIE_DE_DONNEES80 > WHERE PROJET_ID80 = < NUMERO > ;  
UPDATE R_LIAISON_SERIE_DE_DONNEES_PROJET SET  
COMMENTAIRE_LIAISON_SERIE_DE_DONNEES80 = <  
COMMENTAIRE_LIAISON_SERIE_DE_DONNEES80 > WHERE PROJET_ID80 = < NUMERO > ;
```

```
DELETE FROM R_LIAISON_SERIE_DE_DONNEES_PROJET  
WHERE PROJET_ID80 = < NUMERO > ;
```

- **Table R_SOURCE_EXTERNE_PROJET**

```
INSERT INTO R_SOURCE_EXTERNE_PROJET (  
PROJET_ID1,  
PROJET_ID2,  
DATE_LIAISON_PROJET,  
COMMENTAIRE_LIAISON_PROJET  
)  
VALUES (  
< PROJET_ID1 > ,  
< PROJET_ID2 > ,  
"< DATE_LIAISON_PROJET >" ,  
"< COMMENTAIRE_LIAISON_PROJET >"  
);
```

```
UPDATE R_SOURCE_EXTERNE_PROJET SET PROJET_ID1 = < PROJET_ID1 > WHERE PROJET_ID1
= < NUMERO > ;
UPDATE R_SOURCE_EXTERNE_PROJET SET PROJET_ID2 = < PROJET_ID2 > WHERE PROJET_ID1
= < NUMERO > ;
UPDATE R_SOURCE_EXTERNE_PROJET SET DATE_LIAISON_PROJET = < DATE_LIAISON_PROJET
> WHERE PROJET_ID1 = < NUMERO > ;
UPDATE R_SOURCE_EXTERNE_PROJET SET COMMENTAIRE_LIAISON_PROJET = <
COMMENTAIRE_LIAISON_PROJET > WHERE PROJET_ID1 = < NUMERO > ;
```

```
DELETE FROM R_SOURCE_EXTERNE_PROJET
WHERE PROJET_ID1 = < NUMERO > ;
```

- *Table PROJÉT_RPRODUIT_PRODUCTION*

```
INSERT INTO PROJÉT_RPRODUIT_PRODUCTION (
PROJET_ID24,
PRODUCTION_ID24
)
VALUES (
< PROJÉT_ID24 > ,
< PRODUCTION_ID24 >
);
```

```
UPDATE PROJÉT_RPRODUIT_PRODUCTION SET PROJÉT_ID24 = < PROJÉT_ID24 > WHERE
PROJET_ID24 = < NUMERO > ;
UPDATE PROJÉT_RPRODUIT_PRODUCTION SET PRODUCTION_ID24 = < PRODUCTION_ID24 >
WHERE PROJÉT_ID24 = < NUMERO > ;
```


```
DELETE FROM PROJÉT_RPRODUIT_PRODUCTION
WHERE PROJÉT_ID24 = < NUMERO > ;
```

- *Table PROJÉT_RSEFOURNITDE_PRODUCTION*

```
INSERT INTO PROJÉT_RSEFOURNITDE_PRODUCTION (
PROJET_ID21,
PRODUCTION_ID21,
DATE_LIAISON_PRODUCTION21,
COMMENTAIRE_LIAISON_PRODUCTION21
)
VALUES (
< PROJÉT_ID21 > ,
< PRODUCTION_ID21 > ,
"< DATE_LIAISON_PRODUCTION21 >" ,
"< COMMENTAIRE_LIAISON_PRODUCTION21 >"
);
```

```
UPDATE PROJÉT_RSEFOURNITDE_PRODUCTION SET PROJÉT_ID21 = < PROJÉT_ID21 > WHERE
PROJET_ID21 = < NUMERO > ;
UPDATE PROJÉT_RSEFOURNITDE_PRODUCTION SET PRODUCTION_ID21 = < PRODUCTION_ID21
> WHERE PROJÉT_ID21 = < NUMERO > ;
UPDATE PROJÉT_RSEFOURNITDE_PRODUCTION SET DATE_LIAISON_PRODUCTION21 = <
DATE_LIAISON_PRODUCTION21 > WHERE PROJÉT_ID21 = < NUMERO > ;
UPDATE PROJÉT_RSEFOURNITDE_PRODUCTION SET COMMENTAIRE_LIAISON_PRODUCTION21 =
< COMMENTAIRE_LIAISON_PRODUCTION21 > WHERE PROJÉT_ID21 = < NUMERO > ;
```

```
DELETE FROM PROJÉT_RSEFOURNITDE_PRODUCTION
WHERE PROJÉT_ID21 = < NUMERO > ;
```

 suppression de période de projet

(Dans la base de données "échange")

-- Cette requête permet d'enregistrer la demande de suppression du projet dans la base de données "échange".

INSERT INTO SUPPRESSION_ARCHIVAGE

```
(DATE_REC_SUPPRESSION_ARCHIVAGE,TYPE_SUPPRESSION_ARCHIVAGE_ID1,PRENOM_DEMANDEUR,NOM_DEMANDEUR,MAIL_DEMANDEUR,MOTIVATIONS_DEMANDEUR,ID_OBJET_A_SUPPRIMER) VALUES ("SYSDATE" , <id_type_suppression> , "<prenom>" , "<nom>" , "<e_mail>" , "<motivations>" , <num_suppr>);
```

3.3.4. Etude détaillée du C.U. « Consulter une période de projet »

Objectifs :

L'objectif de ce cas d'utilisation est de permettre la consultation de toute une période de projet. On doit y retrouver les caractéristiques, les participants, les productions, la bibliographie, ..., son historique.

Description détaillée :

// L'accès à ce cas d'utilisations peut se faire par de multiples points. Le plus rapide étant de passer par le cadre "menu", puis en cliquant sur "Projets" et enfin, choisir le projet désiré soit dans l'arbre du cadre "menu", soit dans le cadre "contenu".

// Suivant le cas d'utilisation qui lance la "consultation d'une période", l'affichage se fera soit dans le cadre " contenu ", soit dans une nouvelle fenêtre.

Scénario nominal :

1) Tout d'abord le système affiche le nom du projet, le pictogramme correspondant à son état et la date de début de la période.

2) Ensuite, l'affichage de tous les paramètres d'une période va dépendre de son état. Nous pouvons observer ainsi deux cas distincts :

- les projets "*en période d'activité*",
- les projets "*suspendu*" ou "*stoppé*".

Pour les projets "*en période d'activité*" tout doit être afficher : les caractéristiques, les participants, les productions, les données associées, la bibliographie, ...

Pour les autres ("*suspendu*" ou "*stoppé*"), seuls les caractéristiques et l'historique du projet doivent être affichés, car pour les paramètres tel que les productions par exemple, il ne peut pas y avoir de nouvel ajout de production si le projet est suspendu ou stoppé.

3) L'historique du projet est affiché grâce au CU " Consulter l'historique d'un projet ".

// L'utilisateur peut faire toutes les actions qui sont proposées dans le cadre " menu ".

Diagramme de séquence :

Ce CU possède deux particularités :

- Il fait appel à deux bases de données : "archi" et "donnees",
- Il introduit le CU "Consulter l'historique d'un projet".

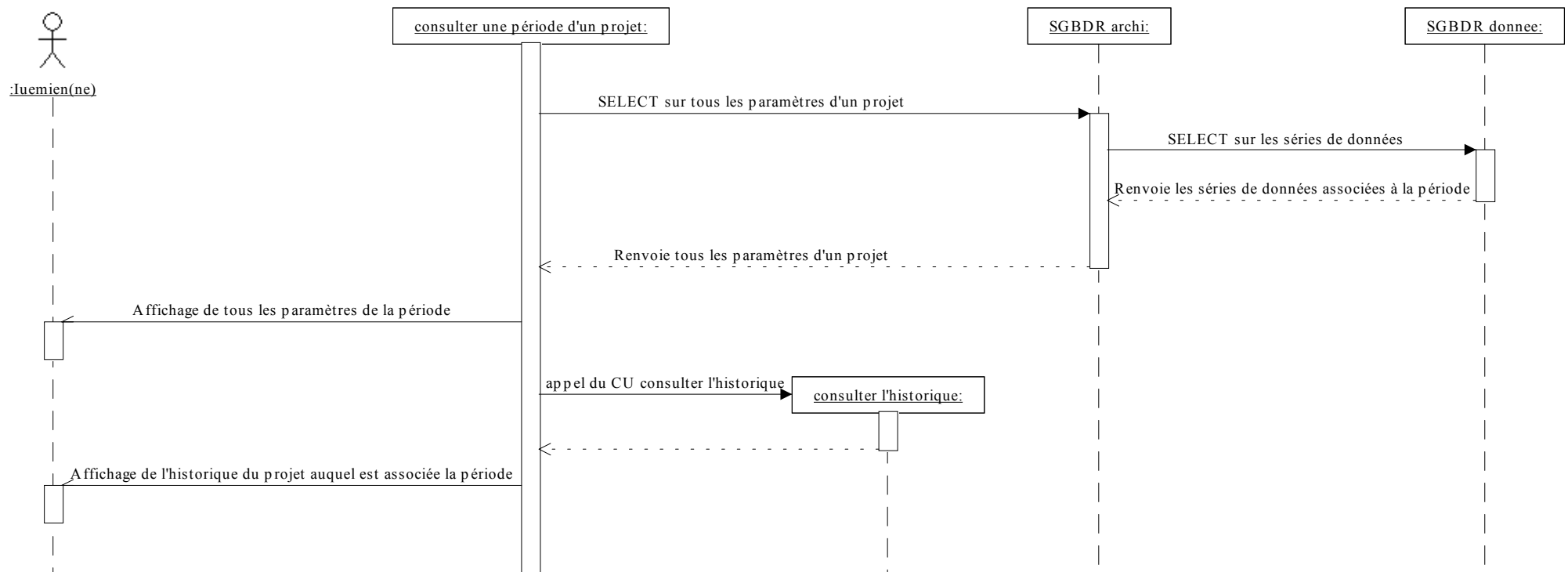


Figure 27 : Diagramme de séquence de "Consulter une période de projet".

Description de l'IHM "Consulter une période de projet" :

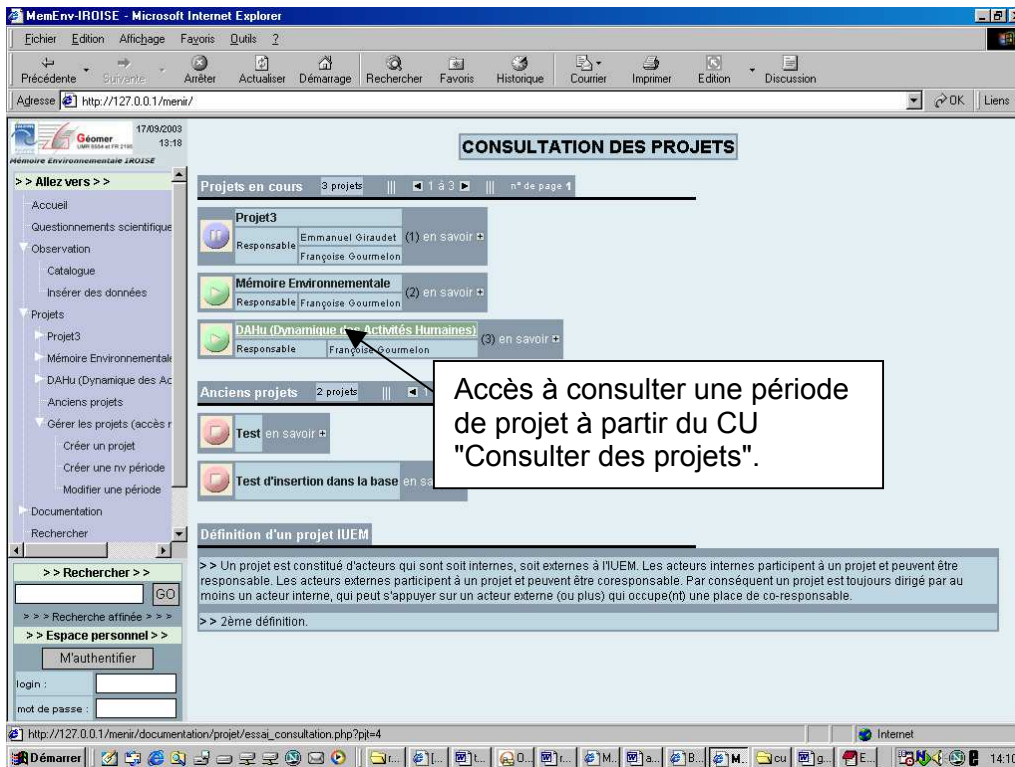


Figure 28 : Copie d'écran de l'IHM "Consulter les projets".

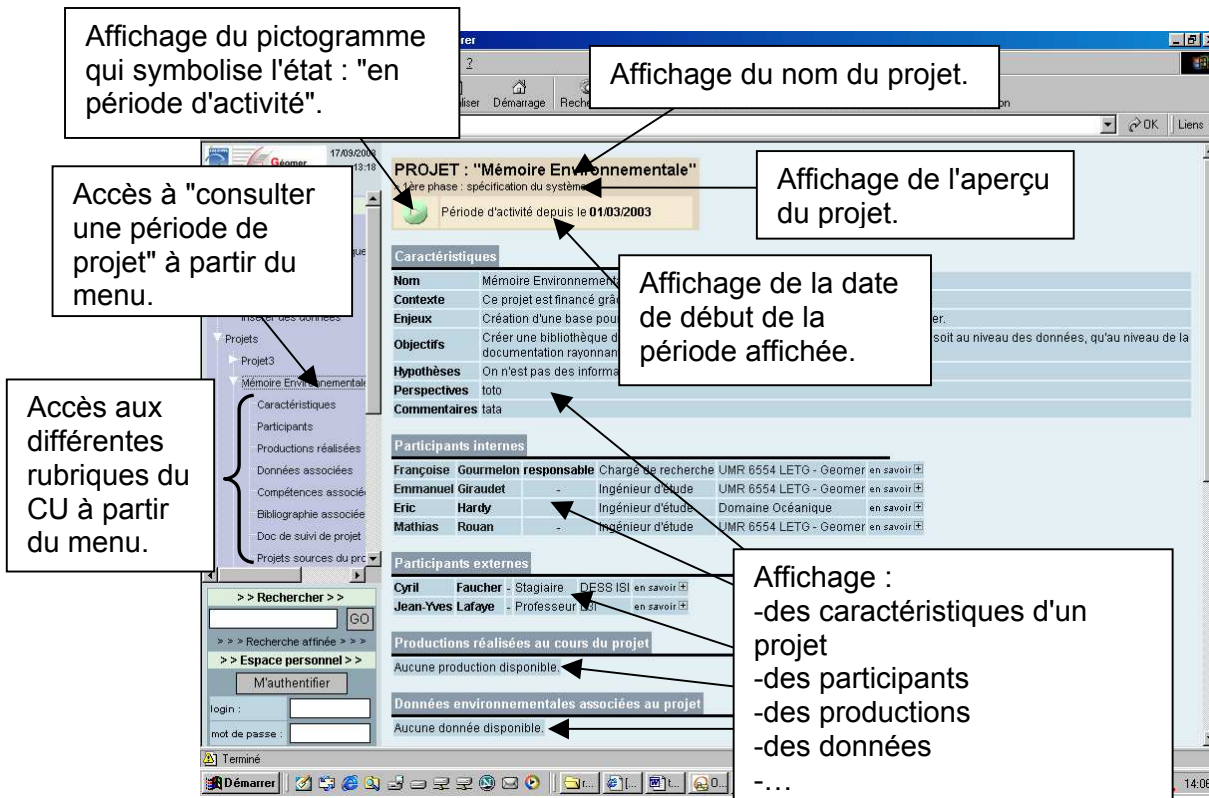


Figure 29 : Copie d'écran de l'IHM "Consulter une période de projet".

(Pour plus de détails sur les différents affichages de ce CU, vous pouvez consulter l'annexe 5 > rubrique tests CU "Consulter une période de projet").

REQUETES SQL du CU "Consulter une période de projet" :

----- -- Affichage des caractéristiques -----

-- Récupération des caractéristiques d'une période de projet

```
SELECT ID_PROJET , NOM_PROJET , CONTEXTE_PROJET , ENJEUX_PROJET ,  
OBJECTIFS_PROJET , HYPOTHESES_PROJET , COMMENTAIRES_PROJET , PROJET_ETAT_ID1 ,  
DATE_ETAT_PROJET , PROJET_SOURCE_PROPRE , APERCU_ETAT_PROJET ,  
PERSPECTIVES_PROJET  
FROM PROJET  
WHERE ID_PROJET = <id de la période affichée>;
```

-- Détermination de la date de fin de la période grâce à la date de début de la période suivante si elle existe

```
SELECT ID_PROJET , DATE_ETAT_PROJET  
FROM PROJET  
WHERE PROJET_SOURCE_PROPRE = <id de la période affichée>;
```

-- Récupération du libellé et de l'URL du pictogramme d'état à afficher

```
SELECT ID_PROJET_ETAT , LIBELLE_PROJET_ETAT , URL_IMAGE_PROJET_ETAT  
FROM PROJET_ETAT  
WHERE ID_PROJET_ETAT = <résultat de PROJET_ETAT_ID1 de la période> ;
```

----- -- Affichage des participants internes -----

-- Récupération des acteurs participant au projet affiché

```
SELECT ACTEUR_INTERNE_IUEM_ID1  
FROM R_PARTICIPANT_INTERNE  
WHERE PROJET_ID5 = <id de la période affichée>;
```

-- Récupération des acteurs responsables du projet affiché

```
SELECT ACTEUR_INTERNE_IUEM_ID2  
FROM R_RESPONSABLE_INTERNE  
WHERE PROJET_ID4 = <id de la période affichée>;
```

-- Récupération des ID_ACTEUR des acteurs internes grâce à la table ACTEUR_INTERNE_IUEM

```
SELECT ACTEUR_ID2 FROM ACTEUR_INTERNE_IUEM  
WHERE ID_ACTEUR_INTERNE_IUEM = <ID_ACTEUR_INTERNE_IUEM récupérés précédemment>;
```

-- Récupération des caractéristiques du participant

```
SELECT NOM_ACTEUR , PRENOM_ACTEUR , ID_ACTEUR FROM ACTEUR  
WHERE ID_ACTEUR = <ID_ACTEUR>;
```

-- Récupération des identifiants que possède un acteur pour ensuite parcourir cet historique

```
SELECT ID_HISTORIQUE_ETAT_ACTEUR  
FROM HISTORIQUE_ETAT_ACTEUR  
WHERE ACTEUR_ID1 = <ID_ACTEUR> ORDER BY DATE_ETAT_ACTEUR;
```

-- Récupération des identifiants des équipes de rattachement, des statuts et les dates de liaison/rupture et ceci pour chaque ID_HISTORIQUE_ETAT_ACTEUR, afin de connaître les équipes auxquelles appartenait l'acteur au moment de la période

```
SELECT DATE_RUPTURE_EQUIPE_DE_RATTACHEMENT ,  
DATE_LIAISON_EQUIPE_DE_RATTACHEMENT , EQUIPE_DE_RATTACHEMENT_ID2 , STATUT_ID1  
FROM R_LIAISON_EQUIPE_DE_RATTACHEMENT
```



```
WHERE HISTORIQUE_ETAT_ACTEUR_ID1= <ID_HISTORIQUE_ETAT_ACTEUR stocké  
précédemment>;
```

```
-- Récupération du libellé du statut voulu
```

```
SELECT ID_STATUT , LIBELLE_STATUT FROM STATUT  
WHERE ID_STATUT = <résultat STATUT_ID1 de la requête précédente>;
```

```
-- Récupération du libellé de l'équipe de rattachement voulue
```

```
SELECT ID_EQUIPE_DE_RATTACHEMENT , LIBELLE_EQUIPE_DE_RATTACHEMENT ,  
URL_EQUIPE_DE_RATTACHEMENT  
FROM EQUIPE_DE_RATTACHEMENT  
WHERE ID_EQUIPE_DE_RATTACHEMENT = <résultat EQUIPE_DE_RATTACHEMENT_ID2 de la  
requête plus haut>;
```

-- Affichage des participants externes

```
-- Récupération des acteurs participant au projet affiché
```

```
SELECT ACTEUR_EXTERNE_IUEM_ID1 FROM R_PARTICIPANT_EXTERNE  
WHERE PROJET_ID7 = <id de la période affichée>;
```

```
-- Récupération des acteurs coresponsables du projet affiché
```

```
SELECT ACTEUR_EXTERNE_IUEM_ID2 FROM R_CORESPONSABLE_EXTERNE  
WHERE PROJET_ID6 = <id de la période affichée>;
```

```
-- Récupération des ID_ACTEUR des acteurs internes grâce à la table ACTEUR_EXTERNE_IUEM
```

```
SELECT ACTEUR_ID3 FROM ACTEUR_EXTERNE_IUEM  
WHERE ID_ACTEUR_EXTERNE_IUEM = <ID_ACTEUR_EXTERNE_IUEM récupérés  
précédemment>;
```

```
-- Récupération des caractéristiques du participant
```

```
SELECT NOM_ACTEUR , PRENOM_ACTEUR , ID_ACTEUR FROM ACTEUR  
WHERE ID_ACTEUR = <ID_ACTEUR>;
```

```
-- Récupération des identifiants que possède un acteur pour ensuite parcourir cet historique
```

```
SELECT ID_HISTORIQUE_ETAT_ACTEUR FROM HISTORIQUE_ETAT_ACTEUR  
WHERE ACTEUR_ID1 = ORDER BY DATE_ETAT_ACTEUR;
```

```
-- Récupération des identifiants des équipes de rattachement, des statuts et les dates de liaison/rupture et  
ceci pour chaque ID_HISTORIQUE_ETAT_ACTEUR, afin de connaître les équipes auxquelles appartenait  
l'acteur au moment de la période
```

```
SELECT DATE_RUPTURE_EQUIPE_DE_RATTACHEMENT ,  
DATE_LIAISON_EQUIPE_DE_RATTACHEMENT , EQUIPE_DE_RATTACHEMENT_ID2 , STATUT_ID1  
FROM R_LIAISON_EQUIPE_DE_RATTACHEMENT  
WHERE HISTORIQUE_ETAT_ACTEUR_ID1= <ID_HISTORIQUE_ETAT_ACTEUR stocké  
précédemment>;
```

```
-- Récupération du libellé du statut voulu
```

```
SELECT ID_STATUT , LIBELLE_STATUT FROM STATUT  
WHERE ID_STATUT = <résultat STATUT_ID1 de la requête précédente>;
```

```
-- Récupération du libellé de l'équipe de rattachement voulue
```

```
SELECT ID_EQUIPE_DE_RATTACHEMENT , LIBELLE_EQUIPE_DE_RATTACHEMENT ,  
URL_EQUIPE_DE_RATTACHEMENT  
FROM EQUIPE_DE_RATTACHEMENT  
WHERE ID_EQUIPE_DE_RATTACHEMENT = <résultat EQUIPE_DE_RATTACHEMENT_ID2 de la  
requête plus haut>;
```

-- Affichage des productions

-- Récupération des instances du projet grâce à une boucle

```
SELECT PROJET_SOURCE_PROPRES FROM PROJET
WHERE ID_PROJET = <identifiant du projet consulté>;
```

-- Récupération des identifiants des productions qui sont de la documentation de suivi de projet

```
SELECT PRODUCTION_ID18 FROM DOCUMENTATION_SUIVI_PROJET;
```

-- Récupération des identifiants des productions produites par les instances du projet en interdisant de sélectionner des identifiants de documentation de suivi de projet.

```
SELECT PRODUCTION_ID24 FROM PROJET_RPRODUIT_PRODUCTION
WHERE (PROJET_ID24 = <identifiant du projet n°1> AND PRODUCTION_ID24 != <résultat de la
requête précédente 1ère ligne> AND ... AND PRODUCTION_ID24 != <résultat de la requête précédente nème
ligne>)
OR (PROJET_ID24 = <identifiant du projet n°n> AND PRODUCTION_ID24 != <résultat de la requête
précédente 1ère ligne> AND ... AND PRODUCTION_ID24 != <résultat de la requête précédente nème
ligne>);
```

-- Récupération des caractéristiques des productions sélectionnés précédemment.

```
SELECT ID_PRODUCTION , TITRE_PRODUCTION , URL_PRODUCTION , DATE_PRODUCTION
FROM PRODUCTION
WHERE ID_PRODUCTION = <résultat de la requête précédente 1ère ligne> OR ... OR
ID_PRODUCTION = <résultat de la requête précédente nème ligne> ORDER BY DATE_PRODUCTION
DESC";
```

-- Affichage des données

-- Récupération des identifiants des données qui sont liées aux projets

```
SELECT SERIE_DE_DONNEES_ID80 , DATE_LIAISON_SERIE_DE_DONNEES80
FROM R_LIAISON_SERIE_DE_DONNEES_PROJET
WHERE PROJET_ID80 = <id du 1er projet> OR ... OR PROJET_ID80 = <id du nème projet>
ORDER BY DATE_LIAISON_SERIE_DE_DONNEES80 DESC;
```

-- Récupération dans la base de données "donnee" des caractéristiques des données

```
SELECT ID_SERIE_DE_DONNEES , LIBELLE_SERIE_DE_DONNEES
FROM SERIE_DE_DONNEES
WHERE ID_SERIE_DE_DONNEES = <résultat de la requête précédente 1ère ligne> OR ... OR
ID_SERIE_DE_DONNEES = <résultat de la requête précédente nème ligne>;
```

-- Affichage des compétences

-- Récupération des identifiants des compétences qui sont liées aux projets.

```
SELECT COMPETENCE_ID2 , DATE_LIAISON_COMPETENCE36
FROM R_LIAISON_COMPETENCE_PROJET
WHERE PROJET_ID36 = <id du 1er projet> OR ... OR PROJET_ID36 = <id du nème projet>
ORDER BY DATE_LIAISON_COMPETENCE36 DESC;
```

-- Récupération des caractéristiques des compétences

```
SELECT ID_COMPETENCE , LIBELLE_COMPETENCE FROM COMPETENCE
WHERE ID_COMPETENCE = <résultat de la requête précédente 1ère ligne> OR ... OR
ID_COMPETENCE = <résultat de la requête précédente nème ligne>;
```

-- Affichage de la Bibliographie

-- Récupération des identifiants des productions considérées comme bibliographie pour les instances de projet.

```
SELECT PRODUCTION_ID21 , DATE_LIAISON_PRODUCTION21
FROM PROJET_RSEFOURNITDE_PRODUCTION
WHERE PROJET_ID21 = <id du 1er projet> OR ... OR PROJET_ID21 = <id du nème projet>
ORDER BY DATE_LIAISON_PRODUCTION21 DESC;
```

-- Récupération des caractéristiques des productions

```
SELECT ID_PRODUCTION, TITRE_PRODUCTION , URL_PRODUCTION
FROM PRODUCTION
WHERE ID_PRODUCTION = <résultat de la requête précédente 1ère ligne> OR ... OR
ID_PRODUCTION = <résultat de la requête précédente nème ligne>;
```

-- Affichage de la documentation de suivi de projet

-- Récupération de toutes les productions qui sont de la documentation de suivie de projet et qui ont été liées aux instances de la chaîne de projet ordonné en ordre décroissant par rapport à la date

```
SELECT PRODUCTION.ID_PRODUCTION , PRODUCTION.TITRE_PRODUCTION ,
PRODUCTION.URL_PRODUCTION , PRODUCTION.DATE_PRODUCTION
FROM PRODUCTION , DOCUMENTATION_SUIVI_PROJET , PROJET_RPRODUIT_PRODUCTION
WHERE (PROJET_RPRODUIT_PRODUCTION.PROJET_ID24 = <projet n°1> AND
PRODUCTION.ID_PRODUCTION = PROJET_RPRODUIT_PRODUCTION.PRODUCTION_ID24 AND
PRODUCTION.ID_PRODUCTION = DOCUMENTATION_SUIVI_PROJET.PRODUCTION_ID18)
OR
...
OR (PROJET_RPRODUIT_PRODUCTION.PROJET_ID24 = <projet n°n> AND
PRODUCTION.ID_PRODUCTION = PROJET_RPRODUIT_PRODUCTION.PRODUCTION_ID24 AND
PRODUCTION.ID_PRODUCTION = DOCUMENTATION_SUIVI_PROJET.PRODUCTION_ID18)
ORDER BY PRODUCTION.DATE_PRODUCTION DESC;
```

-- Affichage des sources de projet externe

-- Récupération des identifiants des projets sources externes pour les instances de projet.

```
SELECT PROJET_ID2 , DATE_LIAISON_PROJET
FROM R_SOURCE_EXTERNE_PROJET
WHERE PROJET_ID1 = <id du 1er projet> OR ... OR PROJET_ID1 = <id du nème projet>
ORDER BY DATE_LIAISON_PROJET DESC;
```

-- Récupération des caractéristiques des projets

```
SELECT ID_PROJET, NOM_PROJET
FROM PROJET
WHERE ID_PROJET = <résultat de la requête précédente 1ère ligne> OR ... OR ID_PROJET =
<résultat de la requête précédente nème ligne>;
```

3.4. Architecture du système et les interactions

3.4.1. Diagramme de déploiement de la maquette

L'architecture choisie pour réaliser la maquette de la "**Mémoire Environnementale**" permet d'avoir un client léger, c'est-à-dire que pour utiliser la "**Mémoire Environnementale**" les utilisateurs ont seulement besoin de s'équiper d'un navigateur internet. Le code se situe sur le serveur d'application, il s'exécute sur ce même serveur. Le service web est assuré par le serveur, le SGBDR est également sur cette machine. Le serveur dvd permet d'archiver les fichiers que compte la "**Mémoire Environnementale**", alors que les données sont elles conservées dans les diverses bases de données. Le personnel accède au service web et donc à l'application grâce au réseau du laboratoire **Géomer**.

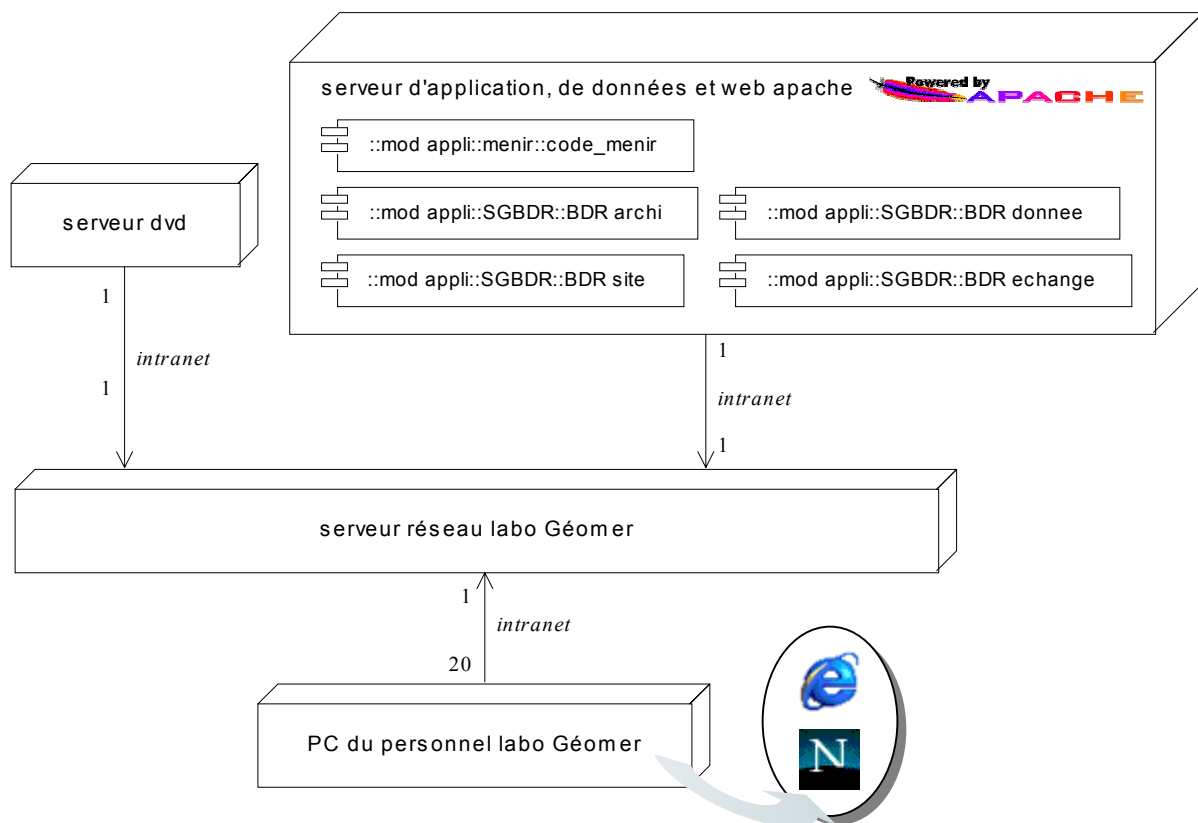


Figure 30 : Diagramme de déploiement de la maquette.

Les nœuds :

- *serveur d'application, de données et web apache :*

Il administre un service web à l'aide d'apache, il administre un système de gestion de bases de données relationnelles, qui est mysql, le code php de l'application gérant la "**Mémoire Environnementale**" y est sauvegardé, il exécute ce code suivant la demande des utilisateurs du service web.

- *serveur dvd* :

Le "serveur dvd" contient les fichiers de la "**Mémoire Environnementale**" : thèses, cartes, ...

- *serveur réseau labo **Géomer*** :

Il administre le réseau du laboratoire **Géomer**.

- *PC du personnel labo **Géomer*** :

Postes sur lesquels travaille le personnel du laboratoire **Géomer**.

Les composants :

- Dans le sous-système "menir" :

- "code_menir" : regroupe le code de l'application : interface, requêtes SQL, images, ...

- Dans le sous-système "SGBDR" :

- "BDR_archi" : base de données dite "*d'archivage*", elle regroupe les données des mémoires "*organisationnelle*", "*globale*" et "*documentaire*". Plus précisément :
 - les acteurs qui participent aux projets et leurs caractéristiques,
 - l'historique des projets,
 - les productions.
- "BDR_donnee" : base de données dite d'observation, elle regroupe les données environnementales. Plus précisément :
 - les catalogues de données,
 - les métadonnées,
 - les données numériques,
 - les liens vers ces données.
- "BDR_site" : base de données qui contient des données relatives à la structure de la maquette comme la composition des barres d'étapes par exemple. Nous retrouvons également à l'intérieur, la définition des principaux objets de la "**Mémoire Environnementale**" comme celle d'un projet ou celle d'une production, le texte affiché à l'accueil, ...
- "BDR_echange" : base de données qui enregistre les demandes de suppression des projets et des productions. Elle pourra être aussi utilisée pour enregistrer les objets auxquels seront abonnés les utilisateurs.

Afin de séparer la partie "*code de l'interface*" de la partie "*Système de Gestion de Bases de Données Relationnelles (SGBDR)*", nous avons créé deux sous-systèmes à l'intérieur du package "*mod appl*". Ces deux sous-systèmes sont : "*menir*", et "*SGBDR*". Ainsi, la conceptualisation de l'implémentation de l'interface est stockée dans le package "*fichiers menir*" du sous-système "*menir*". Le sous-système "*SGBDR*" comprend la conceptualisation de chaque base de données à l'intérieur d'un package propre, par exemple le package "*archi*" contient la structure de la base de données "*archi*".

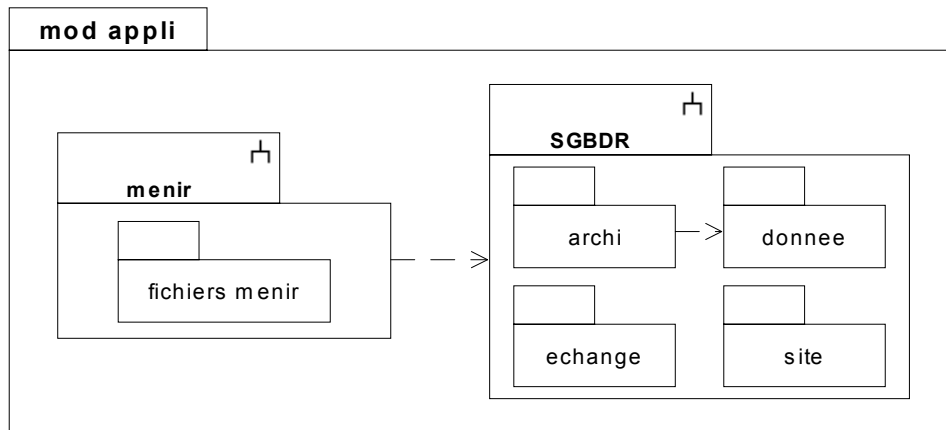


Figure 31 : Diagramme des sous-systèmes.

3.4.2. Interactions entre les acteurs du domaine et le système

Le diagramme de flux qui suit présente les interactions entre les utilisateurs et le système. Nous pouvons remarquer que le sous-système "menir" occupe une place prépondérante dans la réalisation de l'interface entre les acteurs et les bases de données du sous-système "SGBDR". En effet, les quatre acteurs du système interviennent dans ce diagramme et interagissent tous avec "menir". Il constitue la partie "*présentation de l'application*" et par conséquent l'élément avec lequel les acteurs vont interagir directement pour **consulter** ou **insérer** des données. Vous pourrez remarquer que seul "l'administrateur" du système est en relation directe avec les bases de données.

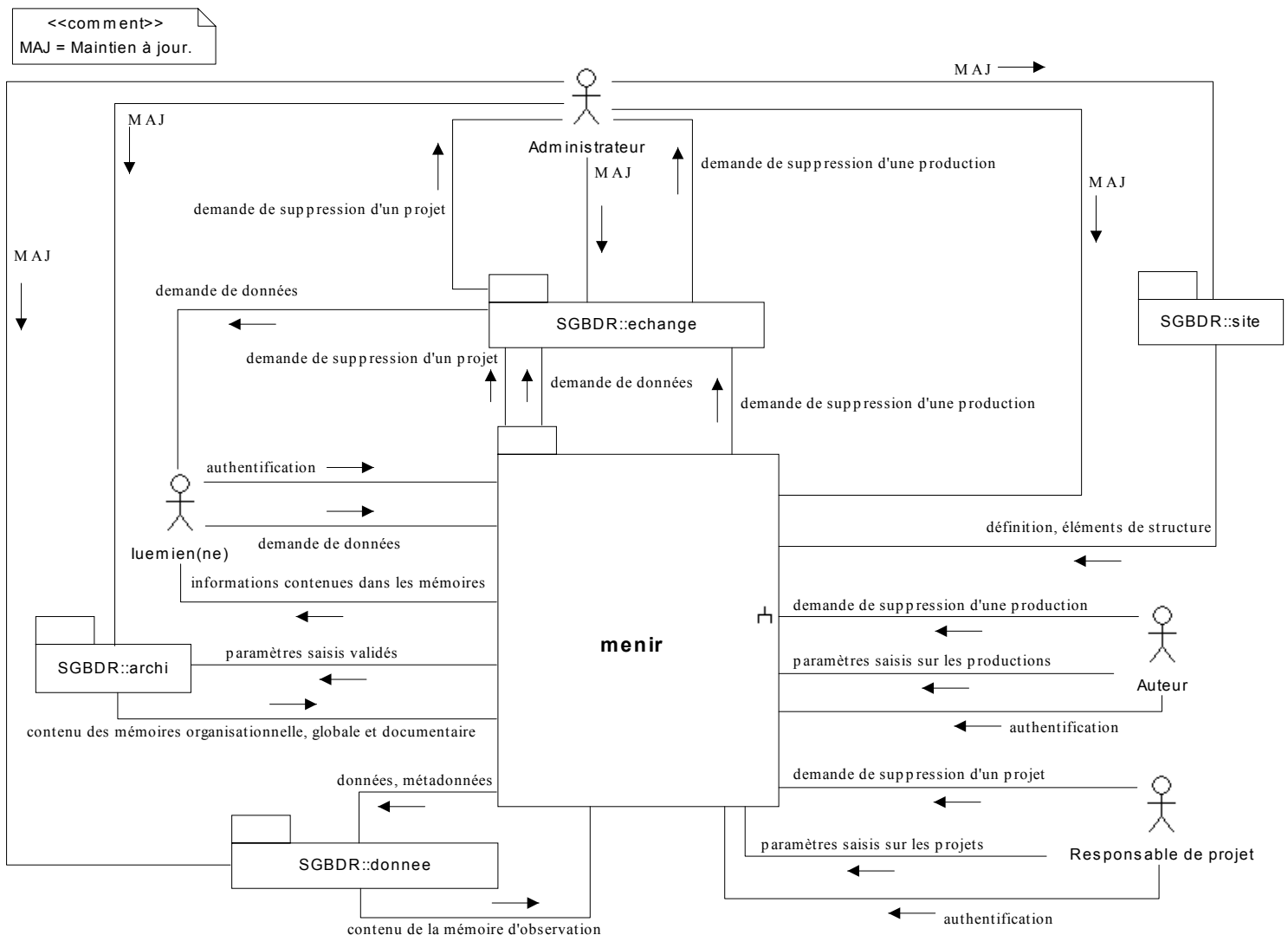


Figure 32 – Diagramme des flux du système dans sa globalité

N.B. : La mise en consultation libre d'informations sur les acteurs devra tout d'abord faire l'objet d'une déclaration à la CNIL.

3.5. Conception de la base de données

3.5.1. L'intégration de la base archi au sein de l'application

La base de données "archi" est intégrée au système de gestion de bases de données (SGBD). Au sein de ce même SGBD, nous retrouvons la base de données intitulée "donnee" dont "archi" dépend. Cette dépendance tient du fait que les tables de relation, qui les lient, sont incluses dans "archi". Ces tables de relations sont par exemple, celles utilisées pour référencer les données associées à un projet ou à une production. Le sens de la dépendance est essentiel dans notre cas, car nous souhaitons que "donnee" reste indépendante et réutilisable au sein d'une autre application et donc que son contenu ne soit pas intimement lié à un élément extérieur.

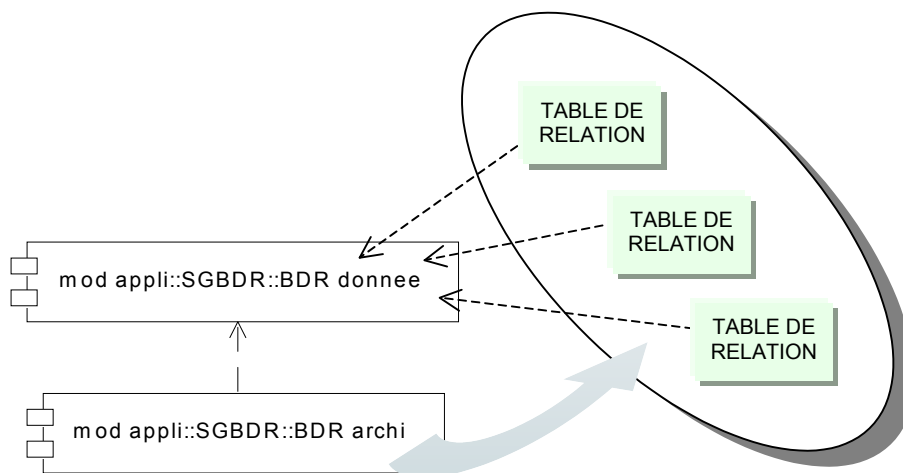


Figure 33 : Diagramme de composants bases de données

3.5.2. Schéma de la base de données relationnelle

Intéressons-nous maintenant à la schématisation de la base de données "archi" à partir du modèle du domaine. Dans ce qui suit seule la schématisation de la "*Mémoire Organisationnelle*" et de la partie "*projet*" de la "*Mémoire Documentaire*" sera abordée, car la spécification de la partie "production" de cette dernière mémoire est en cours d'étude.

Le passage du schéma de données à la base de données relationnelles a été réalisé à l'aide des règles classiques. A noter que les tables de relation sont en vert et que le nom des références (clés étrangères) est formé de la manière suivante :

<ID_><NOM DE LA CLE REFERENCEE><_ID><numéro d'incrémentacion>
exemple : ID_ACTEUR, une de ses clés étrangères est écrite : ACTEUR_ID1.

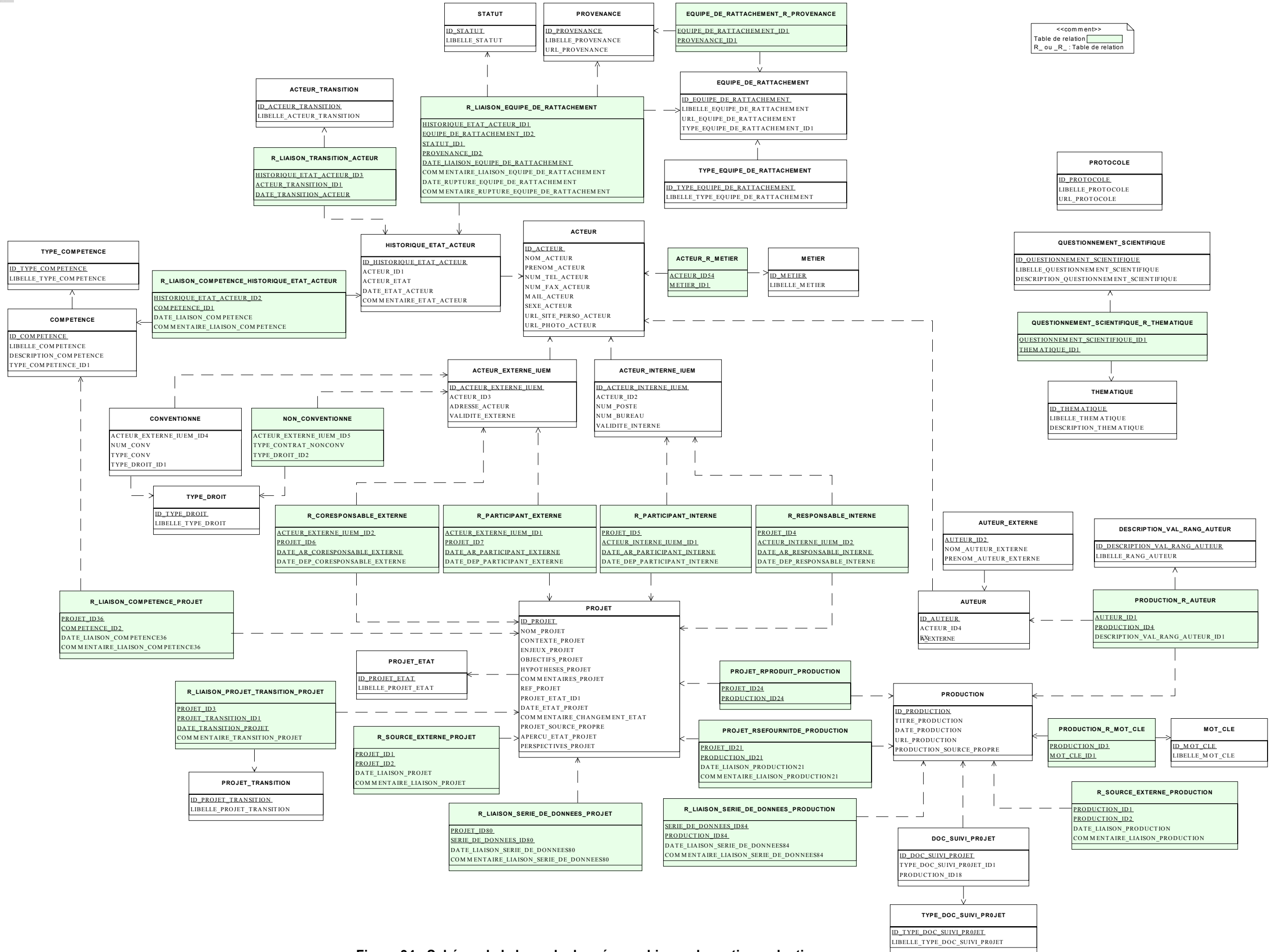


Figure 34 : Schéma de la base de données archi sans la partie production

3.5.3. Description des tables de la base de données "archi"

Les requêtes de création sont faites pour être exécutées à partir du SGBD MySQL, celui-ci ne gère pas le référencement d'un champ par un autre champ et donc ces requêtes ne prennent pas en compte les références (clés étrangères). N.B. : Afin de ne pas alourdir le rapport, la description des tables qui suit ne prend pas en compte toutes les tables de la base de données "archi", la description complète étant en annexe 3.

Création de la base >> MemEnv-IROISE_archi

TABLE : ACTEUR

Champs	Type	Null	Clé	Défaut
<u>ID_ACTEUR</u>	int(11)		PRI	auto_increment
NOM_ACTEUR	varchar(50)			
PRENOM_ACTEUR	varchar(50)			
NUM_TEL_ACTEUR	varchar(24)	YES		
NUM_FAX_ACTEUR	varchar(24)	YES		
MAIL_ACTEUR	varchar(255)	YES	UNI	
URL_SITE_PERSO_ACTEUR	varchar(255)	YES		
URL_PHOTO_ACTEUR	varchar(255)	YES	UNI	
SEXE_ACTEUR	char(1)	YES		

```
CREATE TABLE ACTEUR (
ID_ACTEUR INT NOT NULL AUTO_INCREMENT,
NOM_ACTEUR VARCHAR(50) NOT NULL,
PRENOM_ACTEUR VARCHAR(50) NOT NULL,
NUM_TEL_ACTEUR VARCHAR(24),
NUM_FAX_ACTEUR VARCHAR(24),
MAIL_ACTEUR VARCHAR(255),
URL_SITE_PERSO_ACTEUR VARCHAR(255),
URL_PHOTO_ACTEUR VARCHAR(255),
SEXE_ACTEUR VARCHAR(1),
PRIMARY KEY(ID_ACTEUR),
UNIQUE KEY(MAIL_ACTEUR),
UNIQUE KEY(URL_PHOTO_ACTEUR)
);
```

TABLE : ACTEUR_EXTERNE_IUEM

Champs	Type	Null	Clé	Défaut
<u>ID_ACTEUR_EXTERNE_IUEM</u>	int(11)		PRI	auto_increment
ACTEUR_ID3	int(11)			0 reference(ID_ACTEUR)
ADRESSE_ACTEUR	varchar(255)	YES		
VALIDITE_EXTERNE	int(1)			0

```
CREATE TABLE ACTEUR_EXTERNE_IUEM (
ID_ACTEUR_EXTERNE_IUEM INT NOT NULL AUTO_INCREMENT,
ACTEUR_ID3 INT NOT NULL,
ADRESSE_ACTEUR VARCHAR(255),
VALIDITE_EXTERNE INT NOT NULL.
```

PRIMARY KEY(ID_ACTEUR_EXTERNE_IUEM)
);

TABLE : ACTEUR_INTERNE_IUEM

Champs	Type	Null	Clé	Défaut	
<u>ID_ACTEUR_INTERNE_IUEM</u>	int(11)		PRI		auto_increment
ACTEUR_ID2	int(11)			0	reference(ID_ACTEUR)
NUM_POSTE_ACTEUR	int(4)	YES		0	
NUM_BUREAU_ACTEUR	varchar(10)	YES			
VALIDITE_INTERNE	int(1)			0	

```
CREATE TABLE ACTEUR_INTERNE_IUEM (
ID_ACTEUR_INTERNE_IUEM INT NOT NULL AUTO_INCREMENT,
ACTEUR_ID2 INT NOT NULL,
NUM_POSTE_ACTEUR INT,
NUM_BUREAU_ACTEUR VARCHAR(10),
VALIDITE_INTERNE INT NOT NULL,
PRIMARY KEY(ID_ACTEUR_INTERNE_IUEM)
);
```

TABLE : DOCUMENTATION_SUIVI_PROJET

Champs	Type	Null	Clé	Défaut	
<u>ID_DOCUMENTATION_SUIVI_PROJET</u>	int(11)		PRI		auto_increment
PRODUCTION_ID18	int(11)		UNI	0	reference(ID_PRODUCTI ON)
TYPE_DOCUMENTATION_SUIVI_PROJET_ID1	int(11)			0	reference(ID_TYPE_DO CUMENTATION_SUIVI_ PROJET)

```
CREATE TABLE DOCUMENTATION_SUIVI_PROJET (
ID_DOCUMENTATION_SUIVI_PROJET INT NOT NULL AUTO_INCREMENT,
PRODUCTION_ID18 INT NOT NULL,
TYPE_DOCUMENTATION_SUIVI_PROJET_ID1 INT NOT NULL,
PRIMARY KEY(ID_DOCUMENTATION_SUIVI_PROJET),
UNIQUE KEY(PRODUCTION_ID18)
);
```

TABLE : HISTORIQUE_ETAT_ACTEUR

Champs	Type	Null	Clé	Défaut	
<u>ID_HISTORIQUE_ETAT_ACTEUR</u>	int(11)		PRI		auto_increment
ACTEUR_ID1	int(11)			0	reference(ID_ACTEUR)
ETAT_ACTEUR	int(1)			0	
DATE_ETAT_ACTEUR	varchar(10)				
COMMENTAIRE_ETAT_ACTEUR	longtext				

```
CREATE TABLE HISTORIQUE_ETAT_ACTEUR (
ID_HISTORIQUE_ETAT_ACTEUR INT NOT NULL AUTO_INCREMENT,
ACTEUR_ID1 INT NOT NULL,
ETAT_ACTEUR INT NOT NULL,
DATE_ETAT_ACTEUR VARCHAR(10) NOT NULL,
COMMENTAIRE_ETAT_ACTEUR LONGTEXT NOT NULL,
PRIMARY KEY(ID_HISTORIQUE_ETAT_ACTEUR)
);
```

);

TABLE : R_PARTICIPANT_INTERNE

Champs	Type	Null	Clé	Défaut	
<u>ACTEUR_INTERNE_IUEM_ID1</u>	int(11)		PRI	0	reference(ID_ACTEUR_INTE RNE_IUEM)
<u>PROJET_ID5</u>	int(11)		PRI	0	reference(ID_PROJET)
<u>DATE_AR_PARTICIPANT_INTERNE</u>	varchar(10)		PRI		
<u>DATE_DEP_PARTICIPANT_INTERNE</u>	varchar(10)			0	

```
CREATE TABLE R_PARTICIPANT_INTERNE (
ACTEUR_INTERNE_IUEM_ID1 INT NOT NULL,
PROJET_ID5 INT NOT NULL,
DATE_AR_PARTICIPANT_INTERNE VARCHAR(10) NOT NULL,
DATE_DEP_PARTICIPANT_INTERNE VARCHAR(10) NOT NULL,
PRIMARY KEY(ACTEUR_INTERNE_IUEM_ID1 , PROJET_ID5 , DATE_AR_PARTICIPANT_INTERNE)
);
```

TABLE : PRODUCTION

Champs	Type	Null	Clé	Défaut	
<u>ID_PRODUCTION</u>	int(11)		PRI		auto_increment
TITRE_PRODUCTION	varchar(255)				
DATE_PRODUCTION	varchar(10)				
URL_PRODUCTION	varchar(255)	YES			
PRODUCTION_SOURCE_PROPRE	int(11)			0	reference(ID_PRODUCTION)

```
CREATE TABLE PRODUCTION (
ID_PRODUCTION INT NOT NULL AUTO_INCREMENT,
TITRE_PRODUCTION VARCHAR(255) NOT NULL,
DATE_PRODUCTION VARCHAR(10) NOT NULL,
URL_PRODUCTION VARCHAR(255),
PRODUCTION_SOURCE_PROPRE INT NOT NULL,
PRIMARY KEY(ID_PRODUCTION)
);
```

TABLE : PROJET

Champs	Type	Null	Clé	Défaut	
<u>ID_PROJET</u>	int(11)		PRI		auto_increment
NOM_PROJET	varchar(150)				
CONTEXTE_PROJET	longtext				
ENJEUX_PROJET	longtext				
OBJECTIFS_PROJET	longtext				
HYPOTHESES_PROJET	longtext				
COMMENTAIRES_PROJET	longtext	YES			
REF_PROJET	varchar(20)				
PROJET_ETAT_ID1	int(11)			0	reference(ID PROJE
DATE_ETAT_PROJET	varchar(10)				
COMMENTAIRE_CHANGE_ETAT_PROJET	longtext	YES			

COMMENTAIRE_CHANGE_ETAT_PROJET	longtext	YES		
PROJET_SOURCE_PROPRE	int(11)		0	reference(ID_PROJET)
APERCU_ETAT_PROJET	varchar(64)	YES		
PERSPECTIVES_PROJET	longtext	YES		

```
CREATE TABLE PROJET (
ID_PROJET INT NOT NULL AUTO_INCREMENT,
NOM_PROJET VARCHAR(150) NOT NULL,
CONTEXTE_PROJET LONGTEXT NOT NULL,
ENJEUX_PROJET LONGTEXT NOT NULL,
OBJECTIFS_PROJET LONGTEXT NOT NULL,
HYPOTHESES_PROJET LONGTEXT NOT NULL,
COMMENTAIRES_PROJET LONGTEXT,
REF_PROJET VARCHAR(20) NOT NULL,
PROJET_ETAT_ID1 INT NOT NULL,
DATE_ETAT_PROJET VARCHAR(10) NOT NULL,
COMMENTAIRE_CHANGE_ETAT_PROJET LONGTEXT,
PROJET_SOURCE_PROPRE INT NOT NULL,
APERCU_ETAT_PROJET VARCHAR(64),
PERSPECTIVES_PROJET LONGTEXT,
PRIMARY KEY(ID_PROJET)
);
```

TABLE : PROJET_ETAT

Champs	Type	Null	Clé	Défaut
ID_PROJET_ETAT	int(11)		PRI	auto_increment
LIBELLE_PROJET_ETAT	varchar(255)		UNI	
URL_IMAGE_PROJET_ETAT	varchar(255)	YES		

```
CREATE TABLE PROJET_ETAT (
ID_PROJET_ETAT INT NOT NULL AUTO_INCREMENT,
LIBELLE_PROJET_ETAT VARCHAR(255) NOT NULL,
URL_IMAGE_PROJET_ETAT VARCHAR(255),
PRIMARY KEY(ID_PROJET_ETAT),
UNIQUE KEY(LIBELLE_PROJET_ETAT)
);
```

TABLE : PROJET_RPRODUIT_PRODUCTION

Champs	Type	Null	Clé	Défaut
PROJET_ID24	int(11)		PRI	0 reference(ID_PROJET)
PRODUCTION_ID24	int(11)		PRI	0 reference(ID_PRODUCTION)

```
CREATE TABLE PROJET_RPRODUIT_PRODUCTION (
PROJET_ID24 INT NOT NULL,
PRODUCTION_ID24 INT NOT NULL,
PRIMARY KEY(PROJET_ID24 , PRODUCTION_ID24)
);
```

TABLE : PROJET_RSEFOURNITDE_PRODUCTION

Champs	Type	Null	Clé	Défaut	
<u>PROJET_ID21</u>	int(11)		PRI	0	reference(ID_PROJET)
<u>PRODUCTION_ID21</u>	int(11)		PRI	0	reference(ID_PRODUCTION)
DATE_LIAISON_PRODUCTION21	varchar(10)				
COMMENTAIRE_LIAISON_PRODUCTION21	longtext	YES			

```
CREATE TABLE PROJET_RSEFOURNITDE_PRODUCTION (
PROJET_ID21 INT NOT NULL,
PRODUCTION_ID21 INT NOT NULL,
DATE_LIAISON_PRODUCTION21 VARCHAR(10) NOT NULL,
COMMENTAIRE_LIAISON_PRODUCTION21 LONGTEXT,
PRIMARY KEY(PROJET_ID21 , PRODUCTION_ID21)
);
```

TABLE : PROJET_TRANSITION

Champs	Type	Null	Clé	Défaut	
<u>ID_PROJET_TRANSITION</u>	int(11)		PRI	auto_increment	
LIBELLE_PROJET_TRANSITION	varchar(255)		UNI		

```
CREATE TABLE PROJET_TRANSITION (
ID_PROJET_TRANSITION INT NOT NULL AUTO_INCREMENT,
LIBELLE_PROJET_TRANSITION VARCHAR(255) NOT NULL,
PRIMARY KEY(ID_PROJET_TRANSITION),
UNIQUE KEY(LIBELLE_PROJET_TRANSITION)
);
```

TABLE : R_LIAISON_COMPETENCE_HISTORIQUE_ETAT_ACTEUR

Champs	Type	Null	Clé	Défaut	
<u>HISTORIQUE_ETAT_ACTEUR_ID2</u>	int(11)		PRI	0	reference(ID_HISTORIQUE_ETAT_ACTEUR)
<u>COMPETENCE_ID1</u>	int(11)		PRI	0	reference(ID_COMPETENCE)
DATE_LIAISON_COMPETENCE	varchar(10)				
COMMENTAIRE_LIAISON_COMPETENCE	longtext	YES			

```
CREATE TABLE R_LIAISON_COMPETENCE_HISTORIQUE_ETAT_ACTEUR (
HISTORIQUE_ETAT_ACTEUR_ID2 INT NOT NULL,
COMPETENCE_ID1 INT NOT NULL,
DATE_LIAISON_COMPETENCE VARCHAR(10) NOT NULL,
COMMENTAIRE_LIAISON_COMPETENCE LONGTEXT,
PRIMARY KEY(HISTORIQUE_ETAT_ACTEUR_ID2 , COMPETENCE_ID1)
);
```

TABLE : R_LIAISON_COMPETENCE_PROJET

Champs	Type	Null	Clé	Défaut	
<u>PROJET_ID36</u>	int(11)		PRI	0	reference(ID_PROJET)
<u>COMPETENCE_ID2</u>	int(11)		PRI	0	reference(ID_COMPETENCE)
DATE_LIAISON_COMPETENCE36	varchar(10)				
COMMENTAIRE_LIAISON_COMPETENCE36	longtext	YES			

```
CREATE TABLE R_LIAISON_COMPETENCE_PROJET (
PROJET_ID36 INT NOT NULL,
COMPETENCE_ID2 INT NOT NULL,
DATE_LIAISON_COMPETENCE36 VARCHAR(10) NOT NULL,
COMMENTAIRE_LIAISON_COMPETENCE36 LONGTEXT,
PRIMARY KEY(PROJET_ID36 , COMPETENCE_ID2)
);
```

TABLE : R_LIAISON_PROJET_TRANSITION_PROJET

Champs	Type	Null	Clé	Défaut	
<u>PROJET_ID3</u>	int(11)		PRI	0	reference(ID_PROJET)
<u>PROJET_TRANSITION_ID1</u>	int(11)		PRI	0	reference(ID_PROJET_TRANSITION)
<u>DATE_TRANSITION_PROJET</u>	varchar(10)		PRI		
COMMENTAIRE_TRANSITION_PROJET	longtext				

```
CREATE TABLE R_LIAISON_PROJET_TRANSITION_PROJET (
PROJET_ID3 INT NOT NULL,
PROJET_TRANSITION_ID1 INT NOT NULL,
DATE_TRANSITION_PROJET VARCHAR(10) NOT NULL,
COMMENTAIRE_TRANSITION_PROJET LONGTEXT NOT NULL,
PRIMARY KEY(PROJET_ID3 , PROJET_TRANSITION_ID1 , DATE_TRANSITION_PROJET)
);
```

TABLE : R_LIAISON_SERIE_DE_DONNEES_PRODUCTION

Champs	Type	Null	Clé	Défaut	
<u>PRODUCTION_ID84</u>	int(11)		PRI	0	reference(ID_PRODUCTION)
<u>SERIE_DE_DONNEES_ID84</u>	int(11)		PRI	0	reference(ID_SERIE_DE_DONNEES)
DATE_LIAISON_SERIE_DE_DONNEES84	varchar(10)				
COMMENTAIRE_LIAISON_SERIE_DE_DONNEES84	longtext	YES			

```
CREATE TABLE R_LIAISON_SERIE_DE_DONNEES_PRODUCTION (
PRODUCTION_ID84 INT NOT NULL,
SERIE_DE_DONNEES_ID84 INT NOT NULL,
DATE_LIAISON_SERIE_DE_DONNEES84 VARCHAR(10) NOT NULL,
COMMENTAIRE_LIAISON_SERIE_DE_DONNEES84 LONGTEXT,
PRIMARY KEY(PRODUCTION_ID84 , SERIE_DE_DONNEES_ID84)
);
```

TABLE : R_SOURCE_EXTERNE_PRODUCTION

Champs	Type	Null	Clé	Défaut	
<u>PRODUCTION_ID1</u>	int(11)		PRI	0	reference(ID_PRODUCTI ON)
<u>PRODUCTION_ID2</u>	int(11)		PRI	0	reference(ID_PRODUCTI ON)
DATE_LIAISON_PRODUCTION	varchar(8)				
COMMENTAIRE_LIAISON_PRODUCTION	longtext	YES			

```
CREATE TABLE R_SOURCE_EXTERNE_PRODUCTION (
PRODUCTION_ID1 INT NOT NULL,
PRODUCTION_ID2 INT NOT NULL,
DATE_LIAISON_PRODUCTION VARCHAR(8) NOT NULL,
COMMENTAIRE_LIAISON_PRODUCTION LONGTEXT,
PRIMARY KEY(PRODUCTION_ID1 , PRODUCTION_ID2)
);
```

TABLE : R_SOURCE_EXTERNE_PROJET

Champs	Type	Null	Clé	Défaut	
<u>PROJET_ID1</u>	int(11)		PRI	0	reference(ID_PROJET)
<u>PROJET_ID2</u>	int(11)		PRI	0	reference(ID_PROJET)
DATE_LIAISON_PROJET	varchar(10)				
COMMENTAIRE_LIAISON_PROJET	longtext	YES			

```
CREATE TABLE R_SOURCE_EXTERNE_PROJET (
PROJET_ID1 INT NOT NULL,
PROJET_ID2 INT NOT NULL,
DATE_LIAISON_PROJET VARCHAR(10) NOT NULL,
COMMENTAIRE_LIAISON_PROJET LONGTEXT,
PRIMARY KEY(PROJET_ID1 , PROJET_ID2)
);
```

TABLE : R_RESPONSABLE_INTERNE

Champs	Type	Null	Clé	Défaut	
<u>ACTEUR_INTERNE_IUEM_ID2</u>	int(11)		PRI	0	reference(ID_ACTEUR_IN TERNE_IUEM)
<u>PROJET_ID4</u>	int(11)		PRI	0	reference(ID_PROJET)
<u>DATE_AR_RESPONSABLE_INTERNE</u>	varchar(10)		PRI		
DATE_DEP_RESPONSABLE_INTERNE	varchar(10)			0	

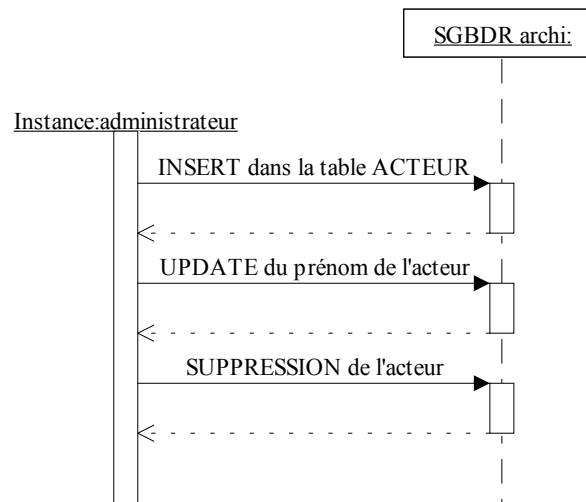
```
CREATE TABLE R_RESPONSABLE_INTERNE (
ACTEUR_INTERNE_IUEM_ID2 INT NOT NULL,
PROJET_ID4 INT NOT NULL,
DATE_AR_RESPONSABLE_INTERNE VARCHAR(10) NOT NULL,
DATE_DEP_RESPONSABLE_INTERNE VARCHAR(10) NOT NULL,
PRIMARY KEY(ACTEUR_INTERNE_IUEM_ID2 , PROJET_ID4 , DATE_AR_RESPONSABLE_INTERNE)
);
```


3.5.4. Scénario de tests de la base de données

Principes

Montrer que les requêtes d'insertion, de modification et de suppression fonctionnent.

Scénario pour la table ACTEUR



Exemples

> **Requête d'insertion :**

INSERT INTO ACTEUR

(NOM_ACTEUR,PRENOM_ACTEUR,NUM_TEL_ACTEUR,NUM_FAX_ACTEUR,MAIL_ACTEUR,
SEXE_ACTEUR)

VALUES (

"nom test" ,
"prenom test" ,
"0202020202" ,
"0303030303" ,
"test@tester.fr" ,
"M"
);

> **Résultat :**

*SELECT * FROM ACTEUR WHERE ID_ACTEUR = 17 ;* affiche :

17	nom test	prenom test	0202020202	0303030303	test@tester.fr	NULL	NULL	M
----	----------	-------------	------------	------------	----------------	------	------	---

> **Requête de modification :**

UPDATE ACTEUR SET PRENOM_ACTEUR = "modif prenom test" WHERE ID_ACTEUR = 17 ;

> **Résultat :**

*SELECT * FROM ACTEUR WHERE ID_ACTEUR = 17 ;* affiche :

17	nom test	modif prenom test	0202020202	0303030303	test@tester.fr	NULL	NULL	M
----	----------	-------------------	------------	------------	----------------	------	------	---

> > > **Requête de suppression :**

DELETE FROM ACTEUR WHERE ID_ACTEUR = 17 ;

Résultat :

Après l'exécution de la requête : `SELECT * FROM ACTEUR WHERE ID_ACTEUR = 17 ;`
Le nombre de réponse = 0.

A noter que les requêtes d'UPDATE, d'INSERT et de DELETE ont été générées pour toutes les tables et l'exemple qui suit vous montre le résultat de cette génération pour la table ACTEUR :

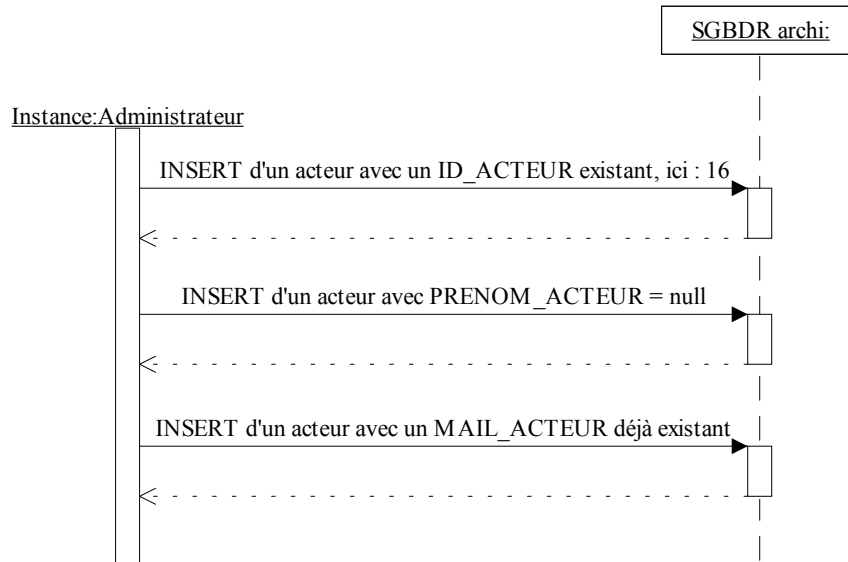
Table > ACTEUR
<code>UPDATE ACTEUR SET ID_ACTEUR = < ID_ACTEUR > WHERE ID_ACTEUR = < NUMERO > ;</code>
<code>UPDATE ACTEUR SET NOM_ACTEUR = < NOM_ACTEUR > WHERE ID_ACTEUR = < NUMERO > ;</code>
<code>UPDATE ACTEUR SET PRENOM_ACTEUR = < PRENOM_ACTEUR > WHERE ID_ACTEUR = < NUMERO > ;</code>
<code>UPDATE ACTEUR SET NUM_TEL_ACTEUR = < NUM_TEL_ACTEUR > WHERE ID_ACTEUR = < NUMERO > ;</code>
<code>UPDATE ACTEUR SET NUM_FAX_ACTEUR = < NUM_FAX_ACTEUR > WHERE ID_ACTEUR = < NUMERO > ;</code>
<code>UPDATE ACTEUR SET MAIL_ACTEUR = < MAIL_ACTEUR > WHERE ID_ACTEUR = < NUMERO > ;</code>
<code>UPDATE ACTEUR SET URL_SITE_PERSO_ACTEUR = < URL_SITE_PERSO_ACTEUR > WHERE ID_ACTEUR = < NUMERO > ;</code>
<code>UPDATE ACTEUR SET URL_PHOTO_ACTEUR = < URL_PHOTO_ACTEUR > WHERE ID_ACTEUR = < NUMERO > ;</code>
<code>UPDATE ACTEUR SET SEXE_ACTEUR = < SEXE_ACTEUR > WHERE ID_ACTEUR = < NUMERO > ;</code>
<code>INSERT INTO ACTEUR (ID_ACTEUR, NOM_ACTEUR, PRENOM_ACTEUR, NUM_TEL_ACTEUR, NUM_FAX_ACTEUR, MAIL_ACTEUR, URL_SITE_PERSO_ACTEUR, URL_PHOTO_ACTEUR, SEXE_ACTEUR) VALUES (< ID_ACTEUR > , "< NOM_ACTEUR >" , "< PRENOM_ACTEUR >" , "< NUM_TEL_ACTEUR >" , "< NUM_FAX_ACTEUR >" , "< MAIL_ACTEUR >" , "< URL_SITE_PERSO_ACTEUR >" , "< URL_PHOTO_ACTEUR >" , "< SEXE_ACTEUR >");</code>
<code>DELETE FROM ACTEUR WHERE ID_ACTEUR = < NUMERO > ;</code>

3.5.5. Tests d'intégrité de la base de données

Principes

Montrer que les contraintes définies lors de la création des tables sont maintenues par MySql

Scénario



Exemples

> **Test de l'unicité d'une clé primaire (ID ACTEUR) : Requête d'insertion avec ID ACTEUR=16, alors que l'enregistrement avec ID ACTEUR=16 existe déjà :**

```
INSERT INTO ACTEUR
(ID_ACTEUR,NOM_ACTEUR,PRENOM_ACTEUR,NUM_TEL_ACTEUR,NUM_FAX_ACTEUR,MAIL_ACTEUR,SEXE_ACTEUR)
VALUES (
16 ,
"nom test" ,
"prenom test" ,
"0202020202" ,
"0303030303" ,
"test@tester.fr" ,
"M"
);
```

> **Résultat :**

"MySQL a répondu:
Duplicata du champ '16' pour la clef 1"

> **Test de la contrainte NOT NULL du champ PRENOM ATEUR : Requête d'insertion PRENOM ACTEUR = null :**

```
INSERT INTO ACTEUR
(NOM_ACTEUR,PRENOM_ACTEUR,NUM_TEL_ACTEUR,NUM_FAX_ACTEUR,MAIL_ACTEUR,SEXE_ACTEUR)
VALUES (
```

```
"nom test" ,  
null ,  
"0202020202" ,  
"0303030303" ,  
"test@tester.fr" ,  
"M"  
);
```

> **Résultat :**

"MySQL a répondu:
Le champ 'PRENOM_ACTEUR' ne peut être vide (null)"

> **Test de l'unicité du champ MAIL ACTEUR : Requête d'insertion pour un mail déjà existant :**

```
INSERT INTO ACTEUR  
(NOM_ACTEUR,PRENOM_ACTEUR,NUM_TEL_ACTEUR,NUM_FAX_ACTEUR,MAIL_ACTEUR,  
SEXE_ACTEUR)  
VALUES (  
"nom test" ,  
"prenom test" ,  
"0202020202" ,  
"0303030303" ,  
"cyril.faucher@orange.fr" ,  
"M"  
);
```

> **Résultat :**

"MySQL a répondu:
Duplicata du champ 'cyril.faucher@orange.fr' pour la clef 3"

3.6. Implémentation de deux cas d'utilisations

Nous avons choisi d'utiliser le langage php pour réaliser les interfaces de la maquette **MEnIr**, car il permet de générer du code HTML, ainsi que d'accéder à des bases de données d'une manière fiable. Au niveau du Système de Gestion de la Base de Données Relationnelles, notre choix c'est porté sur MySql avec EasyPhp, celui-ci constitue une interface de gestion de bases de données MySql implémentée en php. Les contrôles de saisie des formulaires est effectué à l'aide de "fonction javascript" ou par le code php.

3.6.1. Implémentation du C.U. « gérer un projet »

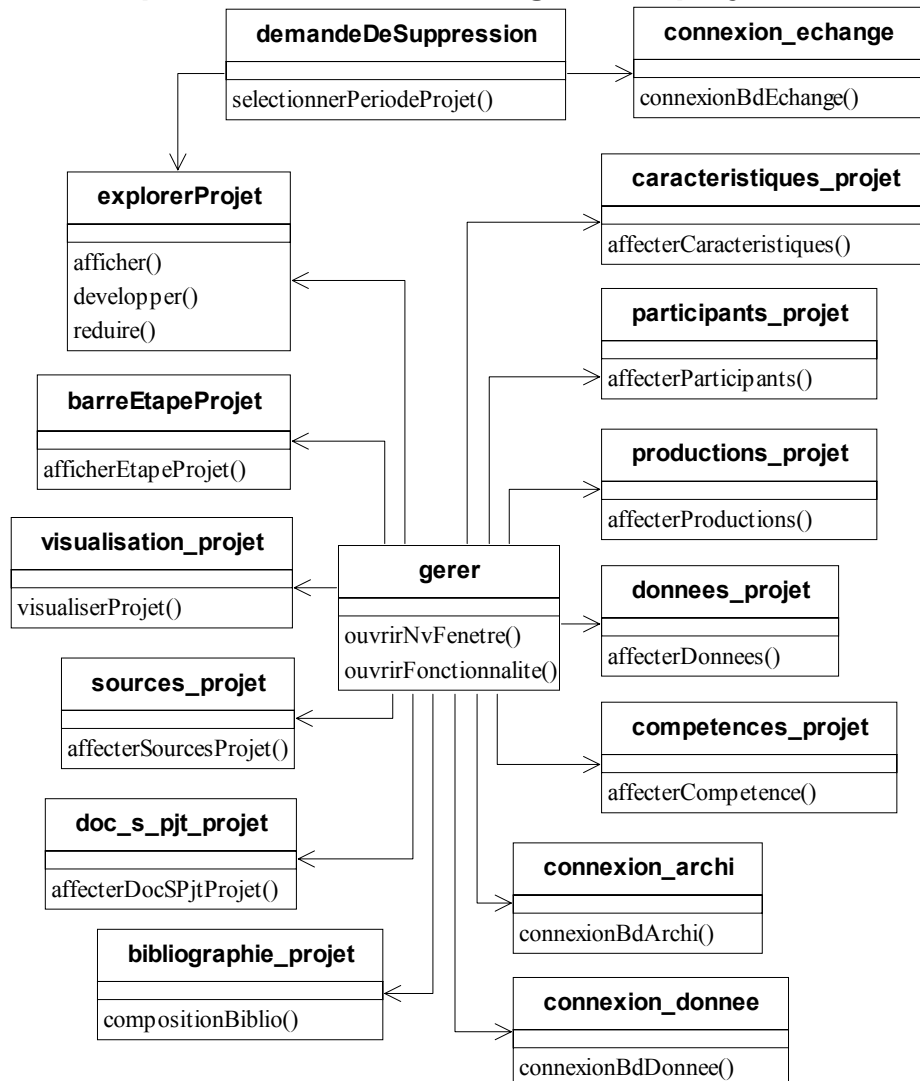
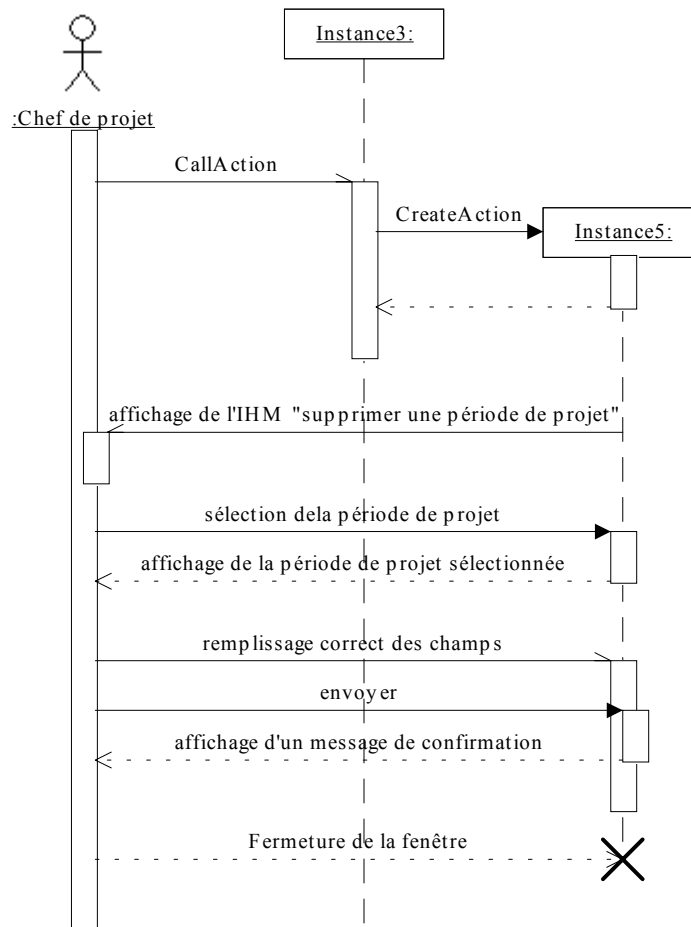


Figure 35 : Schéma de l'implantation du C.U. "Gérer un projet".

Le fichier "gerer" utilise un fichier par chaque partie de l'affichage d'une période de projet comme "productions_projet" et "donnees_projet". Dans chacun de ces fichiers, nous retrouvons une méthode qui permet de saisir ou d'affecter des champs.

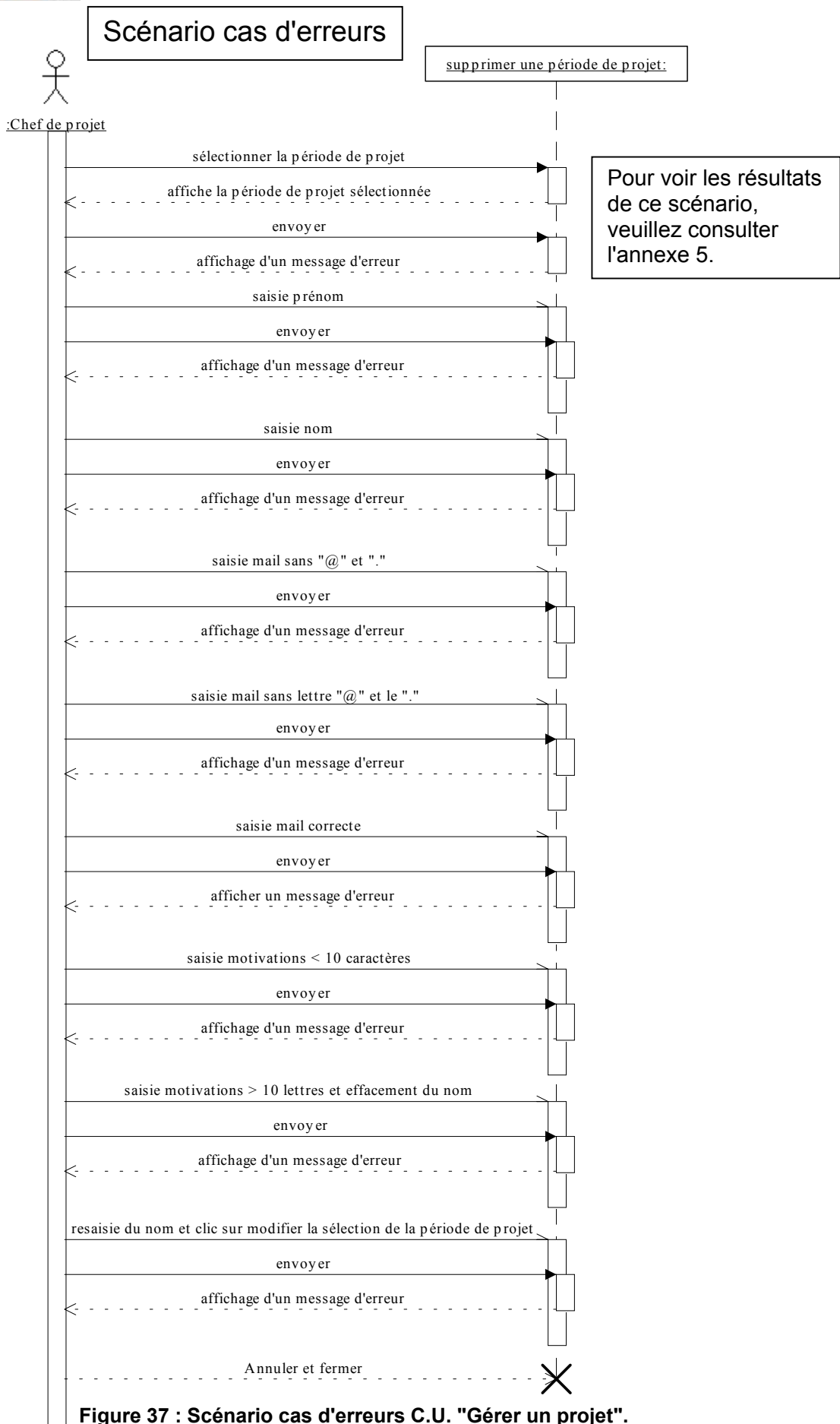
3.6.2. Scénario et tests

Scénario cas nominal



Pour voir les résultats de ce scénario, veuillez consulter l'annexe 5.

Figure 36 Scénario nominal C.U. "Gérer un projet".



3.6.3. Implémentation du C.U. « consulter un projet »

Le fichier "consultation" utilise le fichier "historique_projet" pour afficher l'historique d'un projet. De plus, il utilise deux fichiers de connexion à une base de données : "archi" et "donnee", car à partir du CU "Consulter un période projet" on peut recueillir des informations contenues dans ces deux bases.

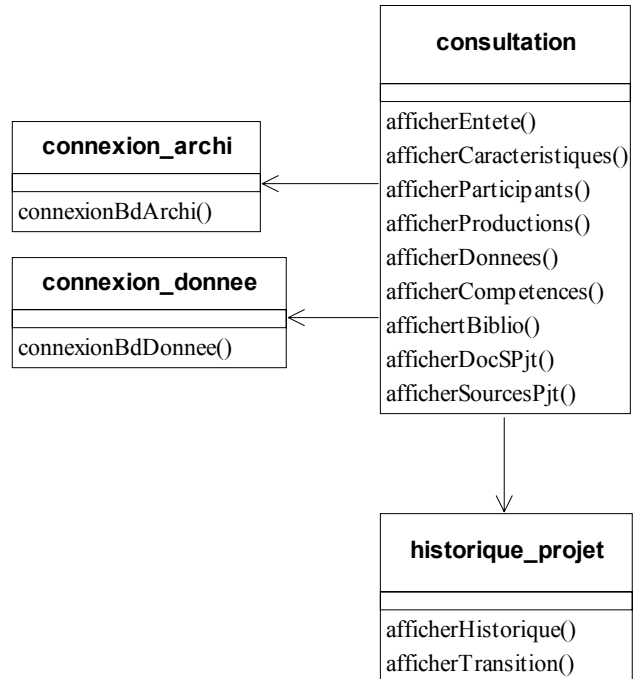
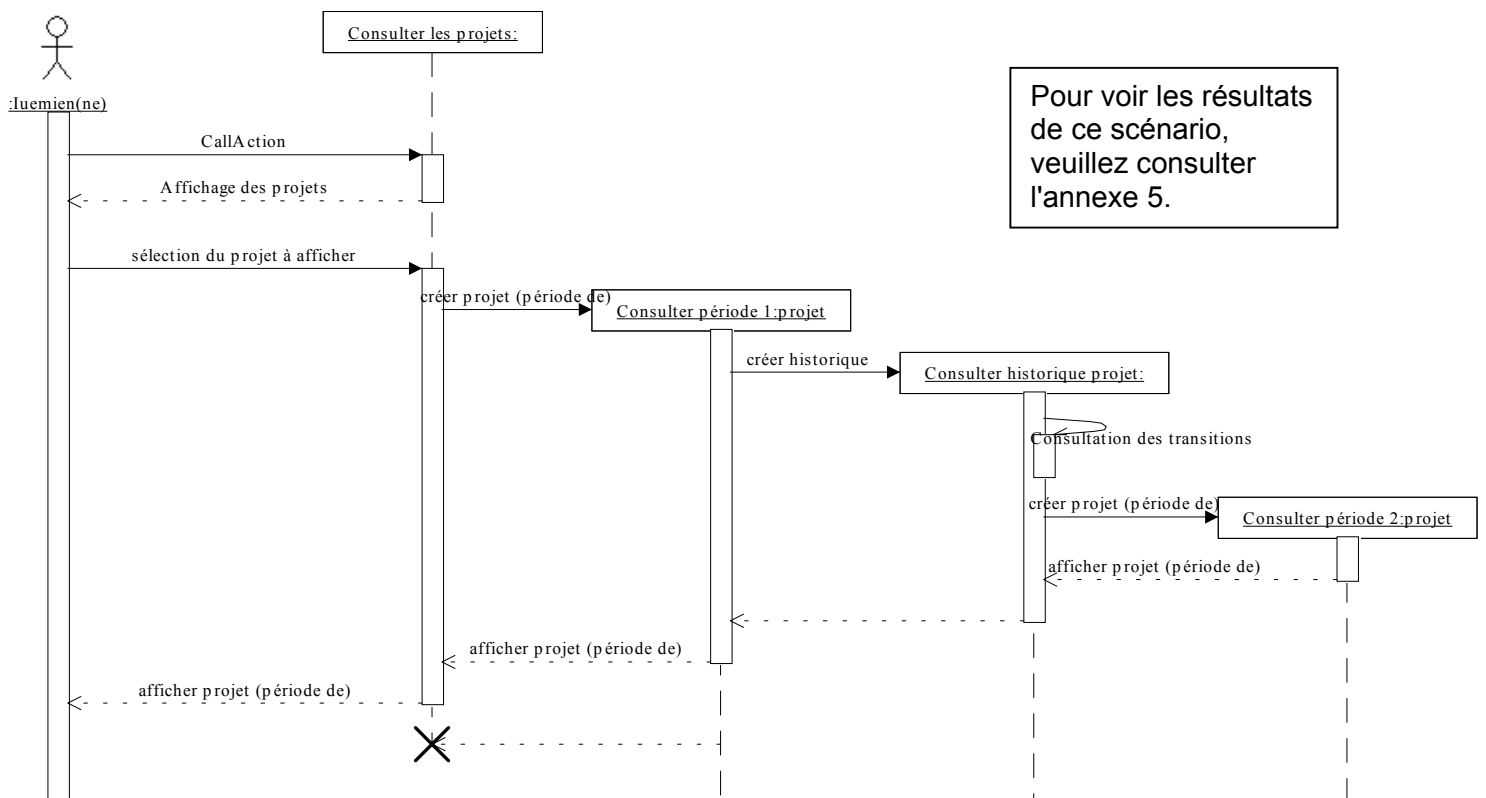


Figure 38 : Schéma de l'implantation du C.U. "Consulter une période de projet".

3.6.4. Scénario et tests



Pour voir les résultats de ce scénario, veuillez consulter l'annexe 5.

Figure 39 : Scénario test C.U. "Consulter une période de projet".

3.7. Bilan du projet

Globalement les objectifs fixés au début du projet ont été atteints, la base de données qui représente la "*Mémoire Organisationnelle*" répond à nos attentes et valide la conceptualisation qui en avait été faite. Les cas d'utilisations implémentés ont validé la structure de la base de données de la "*Mémoire Documentaire*", notamment pour les projets, les participants, les compétences et les équipes de rattachement. L'implémentation de la structure des projets est capitale, car elle fait intervenir de manière forte la notion d'archivage et le concept de chaînage. Pour la "*Mémoire Documentaire*", la spécification proposée dans le diagramme de classes intitulé "*production*" est le fruit d'une première réflexion, celle-ci doit être revue en tenant compte des normes qui existent dans le domaine du référencement bibliographique. Pour cela, nous avons pris contact avec un spécialiste du domaine.

Vous pouvez consulter en *annexe 5* le planning réel des tâches pour avoir une vue globale de la répartition de la charge par semaine.

Au niveau de la conception, quatre cas d'utilisations ont été plus particulièrement traités : "Gérer un projet", "Consulter un projet" (terminé), "Consulter une période de projet" (terminé), "Consulter l'historique d'un projet" (terminé) et dans une moindre mesure : "Consulter acteur" pour lequel la poursuite de l'implémentation va être facilitée par la réutilisation de la plupart des méthodes employées dans le C.U. "Consulter une période de projet". Nous pouvons estimer la production d'environ 6000 lignes de code php, javascript et de requêtes SQL hors création de la base.

Après ces cinq mois de réflexions et d'efforts de conceptualisation, nous pouvons dire que la "**Mémoire Environnementale**" (**MEnIr**) permet de constituer : un archivage des données environnementales de la Mer d'IROISE et un outil de gestion de projet au sein de l'**IUEM**. Plus précisément, **MEnIr** archive les projets de l'**IUEM** et leurs différentes phases, les relations avec les autres projets et avec des productions principalement issues de la recherche scientifique. De plus, **MEnIr** stocke des informations sur les personnes susceptibles d'intervenir dans ces projets. Un lien fort unit ces deux groupes d'informations et tout naturellement les projets vont se relier avec leurs participants. Par la même occasion les données vont se greffer sur cette association en s'appuyant sur de multiples liens. Ces liens sont tous issus des métadonnées et rejoignent les participants, les projets, les productions, ... Tout ceci forme une méta-Mémoire d'informations constituée de trois embryons qui sont : la "*Mémoire Organisationnelle*", la "*Mémoire Documentaire*" et la "*Mémoire d'Observation*". De plus, **MEnIr** se dote d'une quatrième mémoire, cette dernière se veut « globale » et son objectif est de réaliser à un instant donné une vue



générale et synthétique de l'ensemble des informations afin d'étudier l'influence que les objets peuvent avoir entre eux et pourquoi pas d'en déduire des lois d'interactions.

4. CONCLUSION

J'ai pris beaucoup de plaisir à travailler sur le projet **MEnIr**, qui est un projet ambitieux à la fois du point de vue de sa complexité et du point de vue de sa raison d'être. J'ai pu appréhender les notions d'archivage et de conservation de la donnée, qui peuvent paraître simple au premier abord, mais qui s'avèrent complexes lorsque les objets à archiver sont évolutifs.

J'ai pu consolider mes connaissances en modélisation UML et voir l'utilité de ce langage pour un projet d'envergure. J'ai pu apporter mes connaissances en programmation php pour le développement de la maquette.

Le fait de ne pas avoir évolué dans le domaine de ma première compétence m'a demandé de m'adapter au vocabulaire des personnes que j'ai rencontrées et apprendre en quoi consistait leur métier. Ce fut très enrichissant pour ma culture personnelle et fondamental pour la compréhension du sujet.

Dès demain, de nombreuses perspectives sont ouvertes grâce à ce projet, notamment celle d'offrir une garantie de conservation et de suivi des travaux aux différents personnels de **IUEM**. Permettre à tous de valoriser son travail en l'exposant à plus grande échelle. Plus encore, permettre le recoupement de données entre des disciplines différentes pour faire de la pluridisciplinarité un nouveau moteur de recherche.

TABLE DES ILLUSTRATIONS

Figure 1 - Trait de côte de la Mer d'IROISE (source SHOM).	6
Figure 2 : Schéma des packages projets.	12
Figure 3 : Schéma des packages du modèle du domaine.	14
Figure 4 : Chaînage des projets.	17
Figure 5 : DET d'un acteur.	18
Figure 6 : DET d'un projet.	19
Figure 7 : Equivalence d'une classe de type énuméré.	20
Figure 8 – Diagramme de classes décrivant la structure acteur.	22
Figure 9 : Élément "Equipe de rattachement".	23
Figure 10 – Diagramme de classes décrivant la structure de l'historique pour un acteur	24
Figure 11 - Diagramme de classes décrivant le questionnement scientifique.	25
Figure 12 - Diagramme de classes décrivant la structure de la mesure.	26
Figure 13 – Diagramme de classes de la Mémoire Documentaire : Projet.	27
Figure 14 –Diagramme de classes de la Mémoire Documentaire : Production.	28
Figure 15 : Hiérarchie des acteurs.	35
Figure 16 – Cas d'utilisations dits de « gestion ».	36
Figure 17 – Cas d'utilisations dits de « consultation ».	37
Figure 18 : Diagramme de séquence > "Gérer un projet : choix de l'utilisation".	44
Figure 19 : Diagramme de séquence > "Gérer un projet : créer un nouveau projet".	45
Figure 20 : Diagramme de séquence > "Gérer un projet : créer une nouvelle période".	46
Figure 21 : Diagramme de séquence > "Gérer un projet : modifier une période".	47
Figure 22 : Diagramme de séquence > "Gérer un projet : supprimer une période".	48
Figure 23 : Diagramme d'Etat-Transition de "Gérer un projet".	49
Figure 24 : Copie d'écran de l'IHM Gérer les projets pour les événements de 1 à 3.	50
Figure 25 : Copie d'écran de l'IHM Gérer les projets pour l'évènement 4.	50
Figure 26 : Diagramme d'activité de "Gérer un projet".	51
Figure 27 : Diagramme de séquence de "Consulter une période de projet".	60
Figure 28 : Copie d'écran de l'IHM "Consulter les projets".	61
Figure 29 : Copie d'écran de l'IHM "Consulter une période de projet".	61
Figure 30 : Diagramme de déploiement de la maquette.	66
Figure 31 : Diagramme des sous-systèmes.	68
Figure 32 – Diagramme des flux du système dans sa globalité	69
Figure 33 : Diagramme de composants bases de données	70
Figure 34 : Schéma de la base de données archi sans la partie production	71
Figure 35 : Schéma de l'implantation du C.U. "Gérer un projet".	83
Figure 36 Scénario nominal C.U. "Gérer un projet".	84
Figure 37 : Scénario cas d'erreurs C.U. "Gérer un projet".	85
Figure 38 : Schéma de l'implantation du C.U. "Consulter une période de projet".	86
Figure 39 : Scénario test C.U. "Consulter une période de projet".	86

BIBLIOGRAPHIE

Livres :

- P. Roques, F. Vallée, UML en action, 2003, Eyrolles
- A. Wilfred, M. Gupta, K. Bhatnagar, PHP, 2002, CampusPress

Sites internet :

- www.php.net
- www.nexen.net
- www.easyphp.org
- www.laltruiste.com

Normes internationales :

- ISO 19115
- Z39.50
- ISO 2709

GLOSSAIRE

UBO	Université de Bretagne Occidentale.
CNRS	Centre National de la Recherche Scientifique.
UMR	Unité Mixte de Recherche (personnel CNRS et enseignant).
FR	Fédération de recherche.
UML	Unified Modeling Language.
Métadonnée	Fiche qui rassemble les paramètres qui décrivent la donnée : comme le contexte climatique, géographique, la date, l'auteur.
Package	Ensemble qui regroupe des entités de même sémantique ou de fonctionnalités équivalentes.
Classe	Ensemble d'attributs qui définit un objet.
Attribut	Un attribut est une propriété nommée d'une classe qui décrit un domaine de valeurs possibles partagé par tous les objets de la classe.
Table	La table en base de données est la dérivée de la classe en spécification.
Champs	Le champs en base de données est le dérivé de l'attribut en spécification.
Administrateur	Informaticien qui gère un système informatique : application, réseaux.



Les mots-clés :

- > IUEM
- > MEMOIRE
- > PROJET
- > DONNEES
- > SPECIFICATION
- > BASES DE DONNEES RELATIONNELLES

011010111

> **Organisation**

001011100

010101100

> **Documentation**

100100101

101111010

> **Observation**

010101010

101010001

010111010

011110101

010101101

010101010

110111101

110001010

101011100

Mémoire
Environnementale
IROISE



Cyril FAUCHER

LABORATOIRE GEOMER
IUEM 29280 PLOUZANE

DESS ISI - Ingénierie des Systèmes Informatiques -
UNIVERSITE DE LA ROCHELLE

22 avril - 19 septembre 2003

