



# Apport des nouvelles technologies interactives pour l'analyse intégrée en génie électrique : vers un laboratoire virtuel d'expérimentation en électrotechnique

Alita Dewi

## ► To cite this version:

Alita Dewi. Apport des nouvelles technologies interactives pour l'analyse intégrée en génie électrique : vers un laboratoire virtuel d'expérimentation en électrotechnique. Sciences de l'ingénieur [physics]. Institut National Polytechnique de Grenoble - INPG, 2001. Français. NNT : . tel-00597707

**HAL Id: tel-00597707**

**<https://theses.hal.science/tel-00597707>**

Submitted on 1 Jun 2011

**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.

N° attribué par la bibliothèque

///

**THESE**

pour obtenir le grade de

DOCTEUR DE L'INPG

**Spécialité : Génie Electrique**

---

préparée au Laboratoire Electrotechnique de Grenoble  
dans le cadre de l'Ecole Doctorale EEATS

présentée et soutenue publiquement

par

**Alita Dewi**

le 9 juillet 2001

**Titre :**

# Apport des Nouvelles Technologies Interactives pour l'Analyse Intégrée en Génie Électrique : vers un Laboratoire Virtuel d'Expérimentation en Électrotechnique

**Directeur de thèse :**

Jean Louis COULOMB

## JURY

M. André NICOLET  
M. Laurent NICOLAS  
M. Jean-Louis COULOMB  
M. Roland PACAUT  
M. Jean-Pierre DUCREUX  
M. Frédéric DORSCHNER

, Président et Rapporteur  
, Rapporteur  
, Examineur  
, Examineur  
, Examineur  
, Examineur



## Remerciement

J'adresse mes sincères remerciements, tout d'abord à Monsieur Jean-Louis COULOMB, Professeur à l'Institut National Polytechnique de Grenoble - INPG, et Monsieur Roland PACAUT, Maître de Conférences à l'Institut National Polytechnique de Grenoble, pour tous les conseils toujours judicieux qu'ils ont pu prodiguer et pour avoir dirigé mes travaux de thèse. Je conserverai un très bon souvenir de nos discussions, de leur collaboration et de leur façon de toujours m'avoir encouragé pendant ces années de recherche au labo.

Je remercie également :

Monsieur André NICOLET, Professeur à l'Université d'Aix – Marseille III, qui a accepté de cumuler la charge de président et de rapporteur.

Monsieur Laurent NICOLAS, Professeur au CEGELY - Ecole Centrale de Lyon, qui a également accepté d'être rapporteur de ce mémoire.

Monsieur Frédéric DORSCHNER, Ingénieur au Schneider Electric, et Monsieur Jean Pierre DUCREUX, Ingénieur à l'Electricité de France, pour l'honneur qu'il m'a fait en acceptant de participer à ce jury.

Monsieur Marcel IVANES, Professeur à l'Institut National Polytechnique de Grenoble et responsable de la Formation Doctoral, pour avoir accepté de remplacer Monsieur Laurent NICOLAS qui, malheureusement, n'a pas pu assister à ma soutenance de thèse.

Mes remerciements sont également adressés à Monsieur Jean Pierre ROGNON et Monsieur Gérard MEUNIER, respectivement le Directeur du Laboratoire d'Electrotechnique de Grenoble - LEG et le Directeur Adjoint du Laboratoire d'Electrotechnique de Grenoble, pour m'avoir accueilli au laboratoire, me permettant ainsi de travailler dans de bonnes conditions morales et matérielles.

Je voudrais aussi remercier tout le personnel (enseignants, chercheurs, techniciens, secrétaires, ...) du LEG, en particulier Patrick EUSTACHE et Patrice LABIE, qui par leurs conseils, leur aide, leur accueil et leur bonne humeur ont contribué à la bonne réalisation de cette thèse et ce mémoire.





Je remercie aux thésards avec qui j'ai eu le plaisir de travail pendant ces quatre années, particulièrement à Singva MA pour sa très chaleureuse amitié.

Je tiens aussi à remercier le gouvernement français qui a soutenu financièrement cette thèse au travers d'une bourse BGF - Bourse du Gouvernement Français organisé par l'Ambassade de France en Indonésie.

Enfin, un grand merci à mes parents, mes sœurs et mon frère pour leur confiance et leur encouragement pendant mes études.



*A mes parents*



## Tables des matières

Introduction.....	1
Chapitre 1 .....	4
Conception Assistée par Ordinateur (CAO) en Génie Electrique.....	4
1.1. Introduction.....	4
1.2. Principes.....	5
1.2.1. Applications .....	7
1.2.2. Spécifications générales.....	7
1.2.3. Paramètres d'entrée et de sortie .....	8
1.2.4. Architecture.....	9
1.2.5. Technologies .....	10
1.3. Evolution .....	11
1.3.1. Les nécessités industrielles.....	11
1.3.2. Une approche des techniques d'expérimentation dans un laboratoire .....	13
1.3.3. L'implémentation des Nouvelles Technologies Interactives.....	14
1.4. Conclusion .....	14
Chapitre 2 .....	15
Réalité Virtuelle.....	15
2.1. Introduction.....	15
2.2. Systèmes de Réalité Virtuelle .....	16
2.2.1. Les capacités des systèmes de Réalité Virtuelle .....	16
2.2.2.Types de systèmes de Réalité Virtuelle .....	17
2.2.3. Objets dans les systèmes de Réalité Virtuelle .....	17
2.3. Technologies d'interface de Réalité Virtuelle.....	17
2.4. Logiciels de Réalité Virtuelle .....	20
2.4.1. VRML.....	21

2.4.2. Java3D.....	21
2.5. Applications .....	22
2.5.1. Technologie de Réalité Virtuelle en Education.....	22
2.5.2. Technologie de Réalité Virtuelle en Sciences .....	22
2.5.3. Technologie de Réalité Virtuelle en Ingénierie.....	23
2.6. Système de Réalité Virtuelle et CAO.....	24
2.7. Visualisation interactive en électromagnétisme.....	24
2.8. Conclusion .....	25
Chapitre 3 .....	26
Modélisation du Laboratoire d'Expérimentation en Electrotechnique.....	26
3.1. Définition.....	26
3.2. Difficultés .....	27
3.3. Méthodologie .....	28
3.3.1. Modélisation de dispositifs électriques .....	31
3.3.1.1. <i>Un exemple : Les composants du contacteur électromécanique.....</i>	33
3.3.1.2. <i>Un exemple : Les comportements physiques du contacteur électromécanique .....</i>	34
3.3.2. Formalisme des objectifs d'expérimentation .....	36
3.3.3. Formalisme des normes d'expérimentation.....	36
3.3.4. Formalisme des protocoles d'expérimentation.....	37
3.4. Conclusion .....	38
Chapitre 4.....	40
Architecture des Systèmes Interactifs .....	40
4.1. Introduction .....	40
4.2. Modèles d'architecture des systèmes interactifs .....	40
4.2.1. Modèles en couches.....	41
4.2.2. Modèles multi-agents.....	42
4.2.3. Modèles mixtes.....	44
4.3. Modèle AMF .....	45
4.3.1. Structure des facettes.....	45
4.3.2. Composants de la communication .....	46
4.3.2.1. <i>Les ports de communication.....</i>	46

4.3.2.2. Les administrateurs de contrôle (les facettes Contrôle) .....	47
4.3.2.3. Le mécanisme de communication .....	50
4.3.3. Application .....	51
4.4. Conclusion .....	52
Chapitre 5 .....	54
Laboratoire Virtuel d'Expérimentation en Electrotechnique (LVEE) .....	54
5.1. Introduction .....	54
5.2. Développement .....	55
5.3. Spécification .....	55
5.4. Architecture .....	56
5.4.1. Structure hiérarchique des agents interactifs .....	56
5.4.2. Fonctionnalités des agents interactifs .....	57
5.4.3. Communication entre les agents interactifs .....	58
5.4.4. Base de données .....	60
5.4.4.1. Normes d'expérimentation .....	60
5.4.4.2. Matériaux .....	60
5.4.5. Le LVEE par rapport aux logiciels de CAO .....	60
5.4.6. Scénario de simulation .....	62
5.5. Visualisation .....	63
5.5.1. Monde virtuel de l'armoire .....	63
5.5.2. Monde virtuel de la salle d'expérimentation .....	64
5.6. Interaction .....	64
5.7. Pilotage des logiciels de CAO .....	65
5.7.1. Outils de communication .....	66
5.7.2. Interfaces des outils de communication .....	66
5.8. Classes des agents interactifs du LVEE .....	66
5.8.1. Composants AMF .....	67
5.8.2. Agents interactifs du LVEE .....	68
5.8.2.1. Structure des classes .....	68
5.8.3. Extension des agents interactifs dans le LVEE .....	69
5.8.4. Distribution des composants de présentation .....	70



5.9. Conclusion .....	74
Chapitre 6 .....	75
Application du Laboratoire Virtuel d'Expérimentation en Electrotechnique (LVEE) .....	75
6.1. Introduction .....	75
6.2. Un exemple : l'expérimentation de contacteur .....	75
6.2.1. Les valeurs des paramètres du contacteur .....	83
6.2.2. Le calcul et la visualisation des comportements physiques .....	85
6.2.3. Les résultats .....	87
6.2.3.1. Numérique .....	87
6.2.3.2. Graphique animé .....	88
6.2.3.3. Courbes .....	89
6.3. Les impacts souhaités .....	91
6.3.1. Pédagogique .....	91
6.3.2. Industriel .....	92
6.4. Conclusions .....	92
Conclusion générale .....	94
Bibliographie .....	97
ANNEXE A : METHODOLOGIE DE LA MODELISATION DE LABORATOIRE D'EXPERIMENTATION EN ELECTROTECHNIQUE .....	102
ANNEXE B : CATEGORIES DE DISPOSITIFS ELECTRIQUES .....	104
ANNEXE C : COMPORTEMENTS PHYSIQUES D'UN CONTACTEUR ELECTROMECHANIQUE .....	105
ANNEXE D : MODELES MATHÉMATIQUES DES COMPORTEMENTS PHYSIQUES D'UN CONTACTEUR ELECTROMECHANIQUE .....	107
ANNEXE E : .....	131
1 . LES CATEGORIES DES OBJECTIFS D'EXPERIMENTATION .....	131
2 . LES COMPORTEMENTS PHYSIQUES DU CONTACTEUR ET LES OBJECTIFS D'EXPERIMENTATION CORRESPONDANTS .....	134
ANNEXE F : CARACTERISTIQUES DES AGENTS INTERACTIFS DU LVEE .....	136
ANNEXE G : .....	139
1. LES CLASSES DE GESTION DES MATERIAUX .....	139

---

2. LES CLASSES DE GESTION DES NORMES D'EXPERIMENTATION .....	142
ANNEXE H : LE PILOTAGE DE LOGICIELS DE SIMULATION.....	147
ANNEXE I : .....	151
1. MODELE OMT DES OBJETS AMF .....	151
2. LES CLASSES DES COMPOSANTS AMF .....	152
ANNEXE J : SCENE GRAPH DES COMPOSANTS PRESENTATIONS DES DISPOSITIS ELECTRIQUES.....	161
ANNEXE K : ELEMENTS DE ROTERS .....	164
1. THEORIE .....	164
2. APPLICATION .....	165
ANNEXE L : METHODE DE CONCEPTION ET CYCLE EN V.....	175
ANNEXE M : LES NOMS DES DONNEES CREEES AU COURS DE L'EXPERIMENTATION (DE CONTACTEUR) DANS LVEE .....	176
ANNEXE N : SCHEMA DU MECANISME DE COMMUNICATION ENTRE DES AGENTS INTERACTIFS DU LVEE .....	182
ANNEXE O : DIRECTIVE DE L'UTILISATION DE LVEE .....	194
ANNEXE P : .....	196
1. LE MECANISME DE CALCUL ET DE VISUALISATION DES COMPORTEMENTS PHYSIQUES D'UN CONTACTEUR ELECTROMECHANIQUE .....	196
2. LES ROLES DES FACETTES PRESENTATION ET ABSTRACTION DE CHAQUE AGENT INTERACTIF DU MONTAGE D'EXPERIMENTATION D'UN CONTACTEUR.....	198



## Liste des figures

Figure I - 1 Processus fondamental de conception.....	4
Figure I - 2 Composition de modèle informatique de dispositifs électriques.....	6
Figure I - 3 Architecture de logiciels de CAO en Génie Electrique .....	10
Figure II - 1 Head Mounted Display.....	18
Figure II - 2 CABIN .....	19
Figure II - 3 Gant .....	19
Figure II - 4 Eye tracking .....	20
Figure III - 1 Processus d'expérimentation.....	27
Figure III - 2 Méthodologie de modélisation du Laboratoire d'Expérimentation en Electrotechnique.....	30
Figure III - 3 Décomposition structurelle de contacteur électromécanique .....	34
Figure III - 4 Schéma de fonctionnement d'un contacteur électromécanique.....	35
Figure III - 5 Formalisation du problème par la méthode des plans d'expériences. ....	38
Figure IV - 1 Les composants du modèle.....	42
Figure IV - 2 La structure d'un système interactif PAC .....	44
Figure IV - 3 Représentation de la relation hiérarchique des agents AMF .....	45
Figure IV - 4 Représentation des ports de communication d'une facette AMF.....	46
Figure IV - 5 Représentation d'un port de communication exporté.....	47
Figure IV - 6 Modélisation d'un administrateur de contrôle.....	48
Figure IV - 7 La succession des primitives de l'application du modèle AMF .....	52

Figure V - 1 Structure hiérarchique des agents interactifs du LVEE.....	57
Figure V - 2 Mécanisme de la communication entre les agents interactifs.....	59
Figure V - 3 Le LVEE par rapport à l'architecture des logiciels de CAO.....	61
Figure V - 4 Présentation visuelle du LVEE .....	63
Figure V - 5 Processus d'expérimentation dans le LVEE.....	65
Figure V - 6 Structure de pilotage des logiciels de CAO.....	66
Figure V - 7 Classes représentant l'agent d'un contacteur avec l'électroaimant en forme de E .....	69
Figure V - 8 Scene graph des composantes Présentation du LVEE .....	71
Figure V - 9 Scene graph du composant Présentation d'un contacteur .....	71
Figure V - 10 Scene graph du composant Présentation d'un contacteur en E.....	72
Figure V - 11 Scene graph du composant Présentation du montage d'expérimentation pour des contacteurs .....	72
Figure V - 12 Scene graph du composant Présentation de montage pour étudier.....	73
Figure V - 13 Scene graph du composant Présentation de l'agent "Cupboard" .....	73
 Figure VI - 1 Développement de la manipulation d'expérimentation : déterminer le dispositif examiné.....	76
Figure VI - 2 Développement de la manipulation d'expérimentation : déterminer l'objectif d'expérimentation.....	77
Figure VI - 3 Développement de la manipulation d'expérimentation : déterminer l'état mécanique du dispositif .....	77
Figure VI - 4 Développement de la manipulation d'expérimentation : déterminer la charge du dispositif ..	78
Figure VI - 5 Mise en œuvre l'expérimentation : définir les paramètres du contacteur.....	79
Figure VI - 6 Mise en œuvre l'expérimentation : message concernant la relation géométrique entre la bobine et le circuit magnétique du contacteur. ....	79
Figure VI - 7 Menu "Matériaux " .....	80
Figure VI - 8 Mise en œuvre l'expérimentation : faire fonctionner le contacteur.....	81
Figure VI - 9 Observation .....	82
Figure VI - 10 Analyse .....	82
Figure VI - 11 Géométrie de l'électroaimant du contacteur examiné.....	83
Figure VI - 12 Schéma du calcul et de la visualisation des comportements physique du dispositif .....	86
Figure VI - 13 Réponses physiques de contacteur .....	89
Figure VI - 14 Courbes : Force et entrefer .....	90
Figure VI - 15 Courbes : Force et courant de la bobine .....	91

Figure D - 1 Contacteur électromécanique .....	107
Figure D - 2 Répulsion due à la striction .....	110
Figure D - 3 Section de circuit magnétique concernée par la spire de Frager.....	111
Figure D - 4 a. Circuit magnétique dans la région de la spire de Frager, b. Diagramme vecteur des flux, des tensions et des courants induits dans la spire de Frager .....	112
Figure D - 5 Diagramme vecteur des forces magnétomotrices de la surface de pôle aimant entourée par les spires de Frager et de celle non entourée. ....	112
Figure D - 6 Un paquet de tôles ( $X \approx 2Na$ ).....	116
Figure D - 7 Courants de Foucault .....	116
Figure D - 8 Schéma équivalent des tôles parcourues par les courants de Foucault .....	117
Figure D - 9 Profil des trajets des courants de Foucault.....	118
Figure D - 10 Diagramme des forces résistantes.....	124
Figure D - 11 Définition de la surface réelle de contacts .....	126
Figure D - 12 Zones dans l'espace inter électrode .....	128
Figure J - 1 Scene graph du composant Présentation du moteur à Courant Continu .....	161
Figure J - 2 Scene graph du composant Présentation du générateur Synchrone .....	161
Figure J - 3 Scene graph du composant Présentation du générateur Asynchrone .....	162
Figure J - 4 Scene graph du composant Présentation d'autotransformateur .....	162
Figure J - 5 Scene graph du composant Présentation de transformateur avec deux bobinages .....	163



## Liste des tableaux

Tableau III - 1 La séquence des activités de chaque étape du processus d'expérimentation.....	31
Tableau IV - 1 Les situations d'activation des démons.....	47
Tableau IV - 2 Les administrateurs de traduction .....	49
Tableau IV - 3 Les administrateurs comportementaux .....	50
Tableau VI - 1 La configuration des valeurs des paramètres du contacteur.....	84
Tableau VI - 2 Réponse en numérique .....	88





# Introduction



## Introduction

La course à l'innovation dans les technologies des dispositifs électriques, se traduit par une forte intégration des fonctions, par une miniaturisation des composants, par une maîtrise de leur cycle de vie en terme de fiabilité et de durée, ... La conception de ces produits s'appuie sur des logiciels de CAO (Conception Assistée par Ordinateur) capables de répondre à ces nouveaux besoins et donc en mesure d'analyser tous les modèles physiques des dispositifs et ceci de manière intégrée. Ces logiciels doivent donc permettre l'analyse multi-physiques et par la force des choses être multi-méthodes, car une seule méthode ne peut pas répondre correctement à l'ensemble des analyses envisageables.

Cette puissance en terme de modélisation s'accompagne aussi d'une complexité en terme d'utilisation. La technique d'interaction homme - machine de ces logiciels de simulation est désormais un paramètre aussi important que la qualité du modèle développé. Elle doit être conviviale afin de faciliter la tâche du concepteur. L'idée directrice de notre travail est de proposer une interaction homme – machine qui s'inspire directement des techniques d'expérimentation des dispositifs électriques. En effet, l'expérimentation dans un laboratoire est considérée comme une méthodologie naturelle ou du moins, comme un standard du métier, pour analyser ou étudier les phénomènes qui interviennent durant la vie des objets réels. Dans un laboratoire, l'expérimentateur est conduit à développer librement ses idées, ses observations étant orientées par les dispositifs examinés et leurs réactions par rapport aux stimulations dont ils sont l'objet. A l'aide de cette approche, nous espérons que l'utilisation des logiciels de CAO en Génie Electrique sera plus naturelle et plus facile à comprendre que l'approche actuelle qui est plus guidée par les méthodes numériques sous-jacentes.

Notre objectif sera donc d'adapter la démarche des techniques d'expérimentation des dispositifs électriques sous la forme d'un système interactif représentant le Laboratoire d'Expérimentation en Electrotechnique. Ce système devra favoriser la compréhension des phénomènes physiques qui interviennent durant le fonctionnement des dispositifs électriques. Il devra proposer une visualisation naturelle. Les modèles des dispositifs devront se comporter de façon adéquate comme dans le monde réel. Une interaction intuitive devra permettre au concepteur d'interagir directement avec les modèles. Dans un tel système, le fonctionnement du dispositif électrique sera évidemment obtenu par simulation numérique.

Le but de notre travail est de développer un système interactif inspiré du concept du Laboratoire d'Expérimentation en Electrotechnique et servant d'interface homme - machine aux logiciels de CAO en Génie Électrique, en charge de la partie simulation. Par la suite, ce système interactif est appelé le **Laboratoire Virtuel d'Expérimentation en Electrotechnique (LVEE)**. Le problème essentiel de notre travail a été de modéliser le Laboratoire d'Expérimentation en Electrotechnique, car il n'existe pas, à notre connaissance, de formalisme bien défini à ce propos. Nous avons également formalisé les informations nécessitées par les logiciels de CAO en Génie Électrique. Dans notre approche, nous prévoyons l'implantation des technologies de Réalité Virtuelle (RV) afin d'obtenir une visualisation naturelle et une interaction intuitive. Dans notre maquette, l'implantation effective des techniques de réalité virtuelle sera limitée puisqu'elle se fera uniquement par l'intermédiaire d'un écran, d'un clavier et d'une souris 2D. Cette limitation, que l'on peut regretter, a deux explications. D'une part, au début de notre étude, nous n'avions pas accès à la technologie de Réalité Virtuelle nécessaire. D'autre part, nous pensions que le développement d'une telle interface homme – machine devait s'appuyer sur des concepts métiers qu'il fallait absolument dégager avant de se lancer dans une quelconque réalisation.

Dans le premier chapitre de ce mémoire, nous étudions les logiciels de CAO en Génie Électrique actuels et la tendance de leur évolution. En particulier, nous soulignerons le fait que la technique d'interaction homme-machine devient aussi importante que la qualité des modèles développés.

Le deuxième chapitre est consacré à l'étude d'une nouvelle technologie interactive : la technologie de Réalité Virtuelle. Les applications de cette technologie dans les domaines des sciences, de l'ingénierie et de l'éducation sont présentées. A l'avenir, elle devrait fournir des contributions importantes dans le développement des systèmes de CAO.

Dans le troisième chapitre, nous proposons une méthodologie de modélisation d'un Laboratoire d'Expérimentation en Electrotechnique. Nous décomposons, d'abord, ses composants selon les entités engagées dans une expérimentation de dispositif électrique. A l'aide de cette décomposition, nous pouvons structurer et formaliser les connaissances contenues dans ce laboratoire. Les caractéristiques de chaque composant du laboratoire nous conduisent à un modèle d'architecture informatique permettant de définir un système constitué par un ensemble de composants autonomes et coopératifs.

Le quatrième chapitre présente une description brève du modèle multi-agents *AMF* (Agent Multi-Facettes) choisi pour l'architecture de notre Laboratoire Virtuel d'Expérimentation en Électrotechnique.

Le cinquième chapitre est consacré à la réalisation informatique du Laboratoire Virtuel d'Expérimentation en Électrotechnique. Nous avons choisi une approche "gestion des données" pour décrire sa position au milieu des logiciels de CAO en Génie Electrique. Pour notre maquette, nous développons le LVEE comme un système de RV non immersif (interaction utilisant un écran, un clavier et une souris 2D).

Enfin le sixième chapitre présente l'utilisation du système interactif pour l'expérimentation virtuelle d'un contacteur électromécanique. Une étude de la force magnétique du contacteur par rapport au courant de la bobine montre l'interfaçage entre le Laboratoire Virtuel d'Expérimentation en Electrotechnique avec le logiciel Flux2D. L'analyse des impacts souhaités dans les domaines pédagogique et industriel termine ce chapitre.



## Chapitre 1

# **Conception Assistée par Ordinateur (CAO) en Génie Electrique**





# Chapitre 1

## Conception Assistée par Ordinateur (CAO) en Génie Electrique

### 1.1. Introduction

La conception d'un produit est une des étapes essentielles dans le processus industriel. La Conception Assistée par Ordinateur (CAO) correspond à l'apport de l'informatique à cette étape. Dans la CAO, comme dans la technique de conception classique, le processus de conception s'effectue en respectant les étapes fondamentales présentées dans la figure I-1. Le processus de conception comporte les activités de spécification, de synthèse, d'analyse, d'évaluation et de présentation [David 71] [Encarnacao 87]. Le modèle à concevoir est déterminé lors de la spécification. Ensuite, il est développé, modifié et raffiné dans l'activité de synthèse. Les valeurs de ses attributs sont déterminées durant l'activité d'analyse. L'activité d'évaluation correspond à la comparaison entre le résultat de l'analyse et la spécification. Cette activité a, en retour, des influences sur l'activité de synthèse. Enfin, le résultat final achevé est présenté comme le modèle optimal. Comme dans la technique de conception classique, dans la CAO, les modèles possibles développés et les règles de l'analyse et de la synthèse sont formalisés.

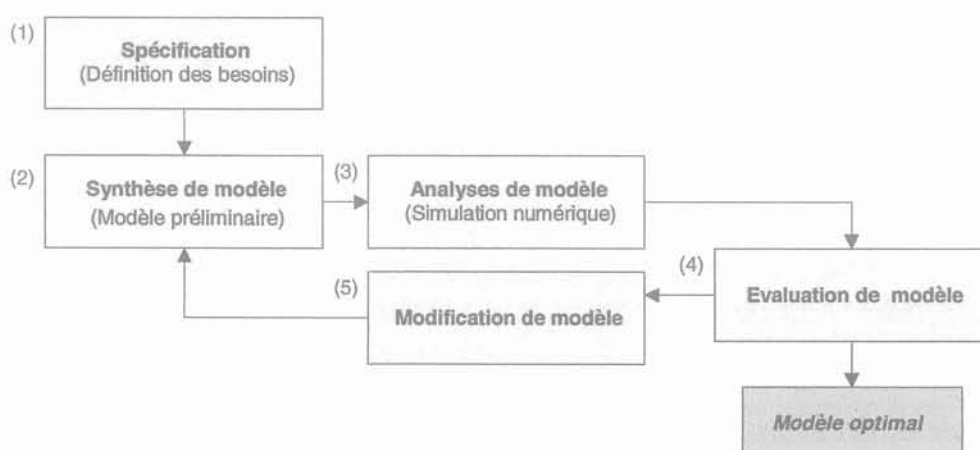


Figure I -1 Processus fondamental de conception

La CAO se définit comme l'intégration des méthodes de l'ingénierie et des sciences informatiques dans un système informatique qui fournit une base des données, une bibliothèque de programmes et un sous système de communication [Barr 85]. En d'autres termes, la CAO est le processus de conception réalisé à l'aide d'un système de CAO, lui-même constitué du logiciel de CAO, du matériel tels que l'écran, l'ordinateur, les outils d'interfaces interactives, etc., et de l'utilisateur. Le logiciel de CAO contient le savoir-faire de la conception. Il est constitué des programmes ou des algorithmes qui permettent le processus de conception dans son ensemble. Étant le cœur du système de CAO, le logiciel de CAO lui-même est souvent assimilé au système de CAO.

L'objectif de la conception assistée par ordinateur est de diminuer le temps de la conception et de faciliter et fiabiliser la manipulation d'un modèle qui soit, si possible, meilleur que le modèle qui serait obtenu sans l'outil informatisé. Pour l'industrie, cette automatisation est essentielle car elle permet d'augmenter le niveau de la productivité.

En Génie Electrique, la CAO a été en premier lieu appliquée à la conception des circuits intégrés des appareils électroniques. Maintenant, elle est utilisée dans toutes ses branches industrielles. Dans ce chapitre, nous discuterons des principes, des applications et des architectures des logiciels de CAO en Génie Electrique. Concernant les évolutions de ces logiciels de CAO, elles tendent à répondre aux besoins de l'industrie du Génie Électrique, c'est-à-dire évoluer vers plus de puissance et, par la force des choses, vers plus de complexité. C'est pour répondre à cette complexité, que nous avons lancé initialement nos travaux. Nous nous sommes efforcés de développer et d'expérimenter un concept qui facilite l'exploitation naturelle des logiciels de CAO. En outre, nous souhaitons que ce concept aide à la compréhension de l'ingénierie et de la physique qui sont évidemment complexes, en fournissant une visualisation interactive et naturelle des phénomènes et des processus physiques mis en jeu [Thollon 99].

## 1.2. Principes

Les logiciels de CAO en Génie Electrique sont utilisés pour construire, sous forme informatique, des modèles ou des prototypes des dispositifs électriques et pour en simuler le fonctionnement et les comportements physiques. Par conséquent, les logiciels de CAO sont aussi appelés logiciels de modélisation, logiciels de simulation ou logiciels d'analyse.

Un modèle de dispositif électrique se compose d'un modèle géométrique et d'un modèle physique (figure I-2). Le modèle géométrique regroupe des paramètres de représentation géométrique alors que le modèle physique contient des paramètres représentant ses propriétés physiques : les lois de comportement des matériaux, les propriétés des sources électriques, etc. ...

Les phénomènes électromagnétiques sont essentiels dans le fonctionnement d'un dispositif électrique. Pour le logiciel de simulation, le problème électromagnétique est le centre de la modélisation

d'un dispositif électrique [Coulomb 85]. Cependant, ces phénomènes sont étroitement liés aux phénomènes thermiques et aux phénomènes mécaniques. Très souvent, les phénomènes thermiques définissent les comportements physiques correspondant aux contraintes principales de la conception. Et très souvent, les performances finales du modèle s'expriment en termes mécaniques.

La simulation de dispositifs électriques correspond à un ou plusieurs problèmes mathématiques. Ainsi, les logiciels de CAO en Génie Electrique contiennent des programmes de calcul ou de simulation numérique, en plus de ceux d'archivage et de présentation graphique. La méthode numérique la plus utilisée, depuis les débuts de la CAO en Génie Electrique, est la méthode des éléments finis. Elle permet de résoudre les systèmes d'équations aux dérivées partielles (équations de Maxwell pour l'électromagnétisme), dans différentes géométries et avec différentes conditions de fonctionnement, en s'appuyant sur une structure de données simple [Hoole 89]. D'autres méthodes numériques généralistes, telles que la méthode des différences finies ou la méthode des intégrales de frontière, sont aussi largement utilisées. Enfin, des méthodes plus spécifiques, telles que la méthode des moments, la méthode des circuits électriques partiels ou la méthode des tubes de flux sont extrêmement utiles lorsqu'elles sont applicables.

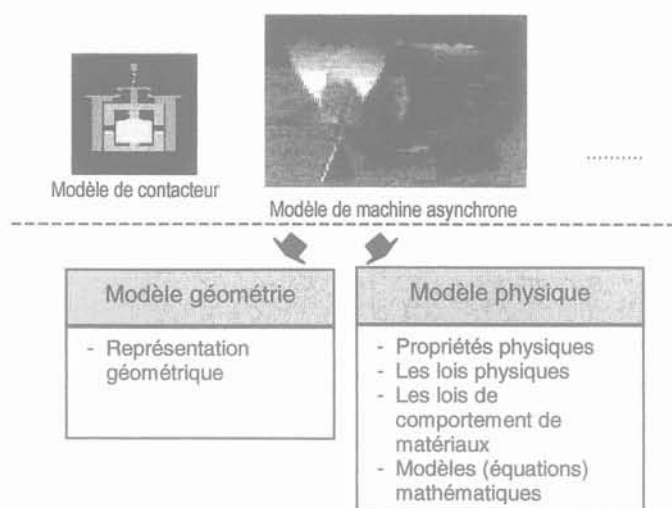


Figure I - 2 Composition de modèle informatique de dispositifs électriques

Les logiciels de CAO en Génie Electrique actuels sont plutôt conditionnés par le modèle numérique dans la mesure où ils sont architecturés autour d'une méthode numérique principale. Avec cette orientation, les modèles peuvent être assez variés en ce qui concerne la géométrie et les propriétés physiques, mais restent dans un moule pour ce qui concerne le modèle physique ou le domaine de la physique. Cependant, la tendance actuelle est à la réalisation d'environnement d'analyse multi-physiques et multi-méthodes [Dular 98] [Maréchal 00]

La diminution du temps de construction de prototypes des dispositifs électriques est l'avantage global apporté par les logiciels de CAO en Génie Electrique. Cet avantage est obtenu par la conjonction de plusieurs services :

- Ils proposent une solution aux problèmes complexes ;
- Ils résolvent ces problèmes complexes très rapidement ;
- Ils fournissent une représentation compréhensible des résultats de la simulation numérique ;
- Ils permettent d'analyser les performances des modèles en cours de conception ;
- ...

### **1.2.1. Applications**

L'industrie est un des domaines dans lequel les logiciels de CAO en Génie Electrique sont appliqués. Un autre domaine d'application est l'université où les logiciels de CAO sont le support de la recherche et de l'éducation. Réciproquement, l'université, par ses recherches, apporte des contributions essentielles au développement des technologies innovantes ainsi que des logiciels de CAO.

Dans le cadre de l'éducation, les logiciels de CAO sont utilisés pour permettre aux étudiants de mieux comprendre les concepts de l'ingénierie présentés dans les cours. Dans ce cas, il est nécessaire que l'utilisation des logiciels de CAO soit aisée et que la visualisation des résultats numériques soit facile à appréhender [Okayama 99].

En définitive, un logiciel de CAO peut être vu comme un outil dans la panoplie du concepteur (c'est sa destination première), comme un objet d'étude et de recherche en soi pour le chercheur ou comme un outil d'aide à la compréhension des phénomènes physiques pour l'enseignement.

### **1.2.2. Spécifications générales**

Pour déterminer les spécifications générales des logiciels de CAO en Génie Electrique, nous partons de son essence même qui repose d'une part, sur une modélisation, et d'autre part, sur une interaction homme - machine.

Les logiciels de CAO en Génie Electrique doivent donc, en premier lieu, permettre de construire un modèle qui prend en compte l'ensemble des phénomènes physiques intervenant durant le fonctionnement du dispositif électrique à concevoir. Or, comme les phénomènes physiques concernés sont nombreux et complexes, la tendance est de spécialiser la résolution de ces problèmes physiques particuliers dans des logiciels indépendants. Par conséquent, pour étudier l'ensemble des problèmes physiques d'un dispositif électrique, le concepteur doit utiliser séparément plusieurs logiciels différents.

En second lieu, la CAO souligne l'importance de l'association de l'homme et de la machine dans une équipe capable de résoudre des problèmes de conception. Les logiciels de CAO en Génie Electrique

doivent donc favoriser une interaction homme - machine qui permette au concepteur de communiquer facilement avec l'ordinateur. Autrement dit, la méthode d'exploration et d'interface homme - machine des logiciels de CAO doivent faciliter les tâches du concepteur. Les logiciels de CAO devraient être capables :

- de faciliter la créativité du concepteur, c'est-à-dire de permettre au concepteur de développer au mieux ses idées ;
- de visualiser les résultats de simulation numérique ou les comportements physiques des dispositifs électriques sous une forme naturelle et facile à interpréter et à analyser.

En résumé, l'ordinateur ne peut que mettre en œuvre l'analyse numérique, il ne possède pas les capacités d'analyse intuitive du concepteur. La technique d'interaction homme - machine est donc essentielle pour un système de CAO.

### **1.2.3. Paramètres d'entrée et de sortie**

Prenons le domaine de la simulation des phénomènes électromagnétiques. Ces phénomènes sont régis par les équations aux dérivées partielles de Maxwell. Par conséquent, les principaux outils informatiques utilisables pour simuler le fonctionnement des dispositifs électriques mettent en œuvre la résolution des équations aux dérivées partielles [Coulomb 85]. Cependant, d'autres approches plus simples sont parfois disponibles. En particulier, lorsque la géométrie et la physique du dispositif modélisé s'y prêtent, des méthodes de type circuit permettent de remplacer les équations aux dérivées partielles par des équations différentielles simples. En fait, cette constatation s'applique aussi aux autres domaines de la physique.

Les paramètres d'entrée des logiciels de CAO en Génie Electrique correspondent aux informations nécessaires pour traiter les équations aux dérivées partielles ou les équations différentielles. Ils comprennent, pour chaque domaine de la physique ou chaque modèle mathématique, des informations sur :

- la géométrie ;
- les caractéristiques des propriétés physiques telles que les matériaux et les sources électriques ;
- les conditions aux limites ;
- la méthode numérique utilisée et ses paramètres, par exemples discrétisation (maillage), nombre d'itérations, précision souhaitée, etc.

En outre, on a aussi besoin, comme paramètres d'entrée, des informations correspondant à l'extraction de résultats de simulation numérique comme, par exemple, l'évolution d'une grandeur ponctuelle ou la valeur d'une intégrale sur une ligne, une surface ou un volume.

Les paramètres de sortie sont les comportements physiques des dispositifs électriques. Les logiciels de CAO en Génie Electrique permettent d'observer essentiellement les comportements physiques correspondant aux phénomènes électromagnétiques : champ magnétique, induction et flux magnétique ; aux performances finales : force magnétique, couple, énergie, etc. et aux contraintes de conception :

température maximale, .... A part ces paramètres de sortie fondamentaux, les logiciels de CAO, suivant modèles physiques fournis, peuvent permettre d'observer d'autres comportements physiques particuliers (trajectoire de particules, ..).

Actuellement, les logiciels de CAO en Génie Electrique permettent la visualisation des paramètres de sortie sous forme graphique en statique ou en dynamique tels que les lignes iso valeurs, les dégradés de couleur d'une grandeur scalaire et les projections d'une grandeur vectorielle.

#### **1.2.4. Architecture**

L'architecture d'un système de CAO particulier dépend des tâches qui doivent être effectuées, des ressources disponibles (hardware et software), de l'expérience de l'équipe de conception du système et des règles en vigueur dans son entreprise [Encarnacao 87]. Nous nous attacherons uniquement au premier de ces critères.

L'architecture du logiciel de CAO en Génie Electrique est donc inspirée par les étapes fondamentales de la conception, comme nous l'avons présenté dans l'introduction, ainsi que par les étapes principales de la simulation :

- la description du problème ;
- la mise en œuvre de la méthode numérique ;
- la vérification, la visualisation, et l'interprétation des résultats.

Nous constatons que les logiciels de CAO en Génie Electrique sont généralement structurés en trois processeurs, qui sont le processeur d'entrée, le processeur du calcul, et le processeur de sortie. Il y a en plus une base des données. Nous décrivons cette structure dans la figure I-3.

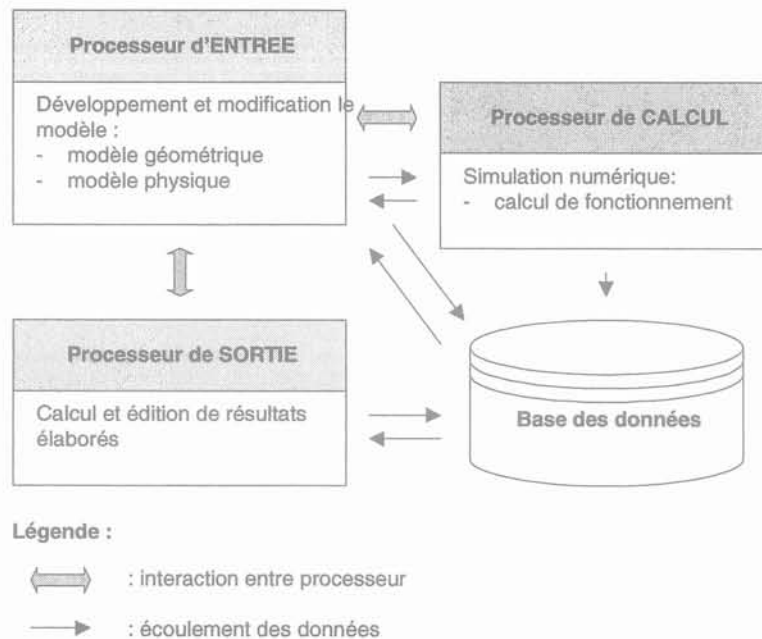


Figure I - 3 Architecture de logiciels de CAO en Génie Electrique

Le processeur d'entrée est chargé de l'activité de synthèse consistant à développer, modifier et raffiner le modèle. Il est constitué des programmes nécessaires pour introduire les informations par des actions du concepteur sur le modèle mis en œuvre. Ces informations peuvent simplement être obtenues par lecture sur fichier ou entrées directement par l'utilisateur à l'aide de l'interface homme - machine. Le processeur d'entrée fonctionne comme un moniteur de communication qui assure les échanges des données et aussi comme un moniteur de conception qui enchaîne les différentes tâches, à la demande du concepteur. Le processeur de calcul contient des codes de simulation numérique. Il est chargé de simuler le fonctionnement des dispositifs électriques. Enfin, le processeur de sortie correspond à la présentation des comportements physiques des dispositifs électriques en traitant les résultats de la simulation numérique.

La base des données gère l'ensemble des informations nécessaires à la bonne conduite de la conception des dispositifs électriques.

### 1.2.5. Technologies

En étudiant la structure des logiciels de CAO en Génie Electrique, nous constatons que sa réalisation prend en compte les trois composantes technologies suivantes :

- l'interface homme – machine ;
- la base des données informatiques ;
- l'architecture logicielle.



Par rapport aux spécifications générales que nous avons décrites précédemment, l'interface homme - machine utilisée devrait faciliter la communication interactive directe entre le concepteur et l'ordinateur dans un environnement conversationnel permettant la synergie homme - machine nécessaire à la conception d'un dispositif électrique. De plus, en fonction de la spécificité du modèle, il est parfois nécessaire que l'interface homme - machine fournisse la technologie de graphique animé [Preis 94]. Elle peut montrer effectivement les différentes positions et formes des composants de dispositifs électriques en fonction du temps. Cela permet d'analyser correctement les performances finales des dispositifs. En effet, un graphique animé permet de vérifier, sur sa maquette virtuelle, si tous les éléments vont jouer correctement leur rôle [Gardan 83].

Cependant, l'implantation de techniques d'animation pose parfois un problème à propos du temps de réponse. D'après Gardan, l'animation qui correspond à l'étude cinématique des différents objets demande la résolution d'équations relativement complexes. Elle peut donc nécessiter un long processus de calcul. D'un autre côté, la rapidité pour assurer une animation en continu est vraiment nécessaire. Dans certain cas, l'animation doit s'effectuer en temps réel, c'est-à-dire que le temps qui s'écoule entre deux images doit être exactement le même que celui qui s'écoulera dans la réalité. Les logiciels de CAO en Génie Electrique apportant des éléments de visualisation qui sont vraiment satisfaisants, ainsi que ceux de simulation numérique, s'appuient ainsi sur l'utilisation d'ordinateurs très puissants ou alors sur des longs calculs préalables.

Concernant la base des données pour un logiciel de CAO, elle doit assurer le stockage, la manipulation, la modification et le transfert en temps réel des données techniques fiables et actualisées sur tous les domaines physiques concernés. En plus, elle devrait permettre de définir le savoir-faire de la conception.

### **1.3. Evolution**

Dans ce sous chapitre, nous parlerons des évolutions des logiciels de CAO en Génie Electrique, en relation avec celles du milieu de l'industrie.

#### **1.3.1. Les nécessités industrielles**

Les nouvelles exigences sur les dispositifs électriques en termes de performances et de fiabilité induisent le besoin d'améliorer les performances des outils logiciels utilisés lors de leur conception [Maréchal 00]. Pour les dispositifs électriques, il est aujourd'hui question d'intégrer leurs fonctions, de les miniaturiser et de maîtriser leur cycle de vie en terme de fiabilité et de durée. Les nouveaux logiciels de CAO concernés devraient donc offrir une simulation prenant en compte plus largement des modèles physiques avec une vision plus globale des dispositifs électriques à simuler.

Autrement dit, le besoin actuel correspond à des logiciels de CAO capables d'analyser tous les modèles physiques et ceci de manière intégrée. Ils doivent donc permettre l'analyse multi-physiques et par la force des choses être multi-méthodes, car une seule méthode ne peut pas répondre correctement à l'ensemble des analyses envisageables.

Une analyse plus fine des besoins conduit aux spécifications suivantes en terme de modélisation :

- *Une modélisation performante;*  
les outils de modélisation (les logiciels de CAO) devraient être capables de résoudre des problèmes concernant plusieurs domaines de la physique et de simuler un dispositif électrique dans son ensemble, ce qui correspond à l'intégration du calcul multi-physiques (exemple : électromagnétisme, thermique, mécanique, ...).
- *Une modélisation diversifiée;*  
les logiciels de CAO devraient favoriser la diversification des problèmes étudiés, non seulement sur la géométrie et le matériau mais aussi sur la physique et les méthodes de résolution (exemple : résolution d'un problème d'électromagnétisme, soit par tubes de flux, soit par éléments finis).
- *Une modélisation intégrant les modèles discontinus;*  
les logiciels de CAO devraient être capables de prendre en compte les discontinuités des modèles physiques et matériaux (exemple : changement de phases solide – liquide).
- *Une modélisation assistée;*  
les logiciels de CAO devraient être complétés par une aide à la modélisation et à la conception pour bénéficier de la puissance évoquée ci-dessus sans se perdre dans sa complexité (exemple : aide contextuelle non seulement opératoire mais aussi adaptée au métier).
- *Une modélisation favorisant l'innovation;*  
les logiciels de CAO devraient permettre le développement et faciliter l'acceptation de nouveaux dispositifs innovants.
- *Une modélisation communicante;*  
les logiciels de CAO, devraient s'intégrer parfaitement dans la chaîne de conception et de fabrication de l'entreprise en favorisant les communications, notamment avec les modelleurs géométriques.
- *Une modélisation inscrite dans les méthodes de travail actuelles;*  
les logiciels de CAO devraient s'inscrire dans les méthodes de travail utilisées en milieu industriel et notamment la co-conception.

En examinant les exigences ci-dessus, nous pouvons déduire que tous les aspects correspondant à la qualité du modèle développé, à la technique de l'interaction homme - machine et l'architecture, sont fortement sollicités.

La *modélisation performante*, la *modélisation diversifiée* et la *modélisation intégrant les modèles discontinus* impliquent que les modèles des dispositifs électriques doivent représenter tous les problèmes physiques possibles correspondants à leurs fonctionnements ainsi que toutes les caractéristiques physiques de leurs composants.

La *modélisation assistée* et la *modélisation favorisant l'innovation* ont des conséquences sur la technique d'interaction homme - machine. Dans la mesure où il est nécessaire d'aborder des problèmes multi-physiques et multi-méthodes, une certaine complexité est à craindre lors de l'utilisation de ces logiciels. Il est nécessaire de prévoir des méthodes qui assistent l'utilisateur, comme par exemple une fonction d'aide. Il serait bon également de développer une méthode d'utilisation proche de standard du métier (§ 1.3.2). De plus, les informations correspondant aux comportements physiques des dispositifs électriques ne sont pas toujours faciles à comprendre par des graphiques statiques ou des animations basiques. Les modèles des dispositifs devraient avoir une visualisation naturelle très proche des dispositifs réels. La visualisation naturelle favorise le processus de validation du modèle. Elle valorise aussi les dispositifs innovants. La technologie de Réalité Virtuelle qui est une des nouvelles technologies de l'interface homme – machine, a été développée afin d'obtenir une visualisation proche de la réalité. Nous parlerons de cette technologie dans le chapitre 2.

Les deux dernières exigences, la *modélisation communicante* et *modélisation inscrite dans les méthodes de travail actuelles*, ont des conséquences sur l'architecture logicielle. Elle doit prendre en compte l'intégration des logiciels de CAO et des solveurs, la distribution des informations et le partage de base des données.

En résumé, ce dont le concepteur a besoin pour simplifier son travail en conception de dispositifs électriques c'est d'un **environnement d'analyse intégrée**, c'est-à-dire un environnement fédérateur capable d'accueillir divers solveurs avec toutes les facilités pour les piloter.

### **1.3.2. Une approche des techniques d'expérimentation dans un laboratoire**

L'expérimentation dans un laboratoire est considérée comme une méthodologie naturelle ou du moins, comme un standard du métier, pour analyser ou étudier les phénomènes qui interviennent durant la vie des objets réels. Dans un laboratoire, l'expérimentateur est conduit à développer librement ses idées, ses observations étant orientées par les objets examinés.

Dans les nouveaux logiciels de CAO en Génie Electrique la technique d'interaction homme - machine, en particulier la méthode d'exploration, est un paramètre important. Pour que l'utilisation du logiciel de CAO soit naturelle et facilite les tâches du concepteur, nous proposons comme méthode d'exploration du logiciel de CAO en Génie Electrique, une approche inspirée directement des techniques d'expérimentation des dispositifs électriques. Le bénéfice est double puisque l'expérience acquise dans le monde réel sera transposable dans le système de CAO et inversement l'expérience obtenue en CAO sera directement transférable sur le dispositif réel.

### 1.3.3. L'implémentation des Nouvelles Technologies Interactives

Pour prendre en compte le standard du métier, l'exploration du logiciel de CAO a besoin de l'implémentation des nouvelles technologies interactives. La technologie de Réalité Virtuelle est une de ces nouvelles technologies. Elle permet au concepteur d'interagir directement avec le modèle développé. Le concepteur peut avoir une interaction intuitive favorisant au mieux sa créativité.

Cette technologie favorise la manipulation d'un modèle virtuel représentant correctement la géométrie et le fonctionnement d'un dispositif. La technologie de Réalité Virtuelle fournit un perfectionnement pour la visualisation d'un système de CAO. Celle-ci améliore la qualité du modèle conçu et du processus d'approbation [ITNews 99].

## 1.4. Conclusion

Après avoir examiné les caractéristiques des logiciels de CAO en Génie Electrique, nous avons souligné qu'ils possèdent deux aspects aussi importants l'un que l'autre : le modèle développé et la technique d'interaction homme - machine [Coulomb 85]. En particulier, un modèle très sophistiqué sera sans intérêt si son utilisation au sein du logiciel de CAO est complexe et longue, et s'il n'existe pas d'assistance aux tâches du concepteur.

Les évolutions dans l'industrie du Génie Electrique s'appuieront sur des nouveaux logiciels de CAO qui devraient permettre d'accomplir l'analyse *multi-physiques* et *multi-méthodes* des dispositifs électriques. La complexité des modèles développés montre que, lors des développements du logiciel de CAO, la recherche d'une interaction homme - machine performante est incontournable, au moins autant que la recherche pour améliorer la qualité des modèles. A la limite, l'utilisation d'un système de CAO idéal devrait être totalement intuitive : l'interface homme - machine travaillerait en totale synchronisation avec notre façon de traiter les informations [Jacobson 94].

Pour se rapprocher d'une méthode d'exploration naturelle, nous proposons pour modéliser, simuler et évaluer un dispositif électrique, une transposition des techniques d'expérimentation dans un Laboratoire d'Expérimentation en Electrotechnique. Cette approche se fera grâce à un système interactif représentant un modèle du Laboratoire d'Expérimentation en Electrotechnique. Nous prévoyons d'utiliser, à terme, une technologie Réalité Virtuelle afin d'avoir une visualisation interactive, naturelle et facile à comprendre. Le système interactif sera développé en considérant qu'il devra s'intégrer dans un système de CAO actuel, c'est-à-dire un environnement d'analyse intégrée en Génie Electrique.



## Chapitre 2

# Réalité Virtuelle



## Chapitre 2

### Réalité Virtuelle

#### 2.1. Introduction

La visualisation numérique est une représentation graphique des données. Elle est réalisée avec l'intention d'aider l'utilisateur à interpréter plus facilement ses données et à en faire ressortir certaines caractéristiques. Depuis l'année 1989, des études sur la technologie de Réalité Virtuelle (RV) sont apparues. L'objectif de ces études est d'obtenir une visualisation naturelle, proche de la réalité.

Le terme de Réalité Virtuelle a été défini dans plusieurs contextes et possède donc des sens différents. D'habitude, ce terme signifie simplement une animation graphique dans un espace 3D. Dans ce cas, il n'est possible que de regarder ou de voyager dans un monde virtuel. En plus de cette première signification, le terme de Réalité Virtuelle est assimilé à la simulation interactive et au prototypage interactif. Dans la simulation interactive, tout en voyageant, l'utilisateur peut interagir avec les objets visualisés. Dans le prototype interactif, l'utilisateur a en plus la possibilité de modifier le comportement de ces objets.

La technologie de RV est donc une technique de représentation permettant de visualiser les informations dans l'espace 3D, de voyager dans cet espace et de manipuler les objets visualisés. Grâce à cette technologie, on peut appréhender les concepts abstraits et les événements de l'échelle atomique à l'échelle planétaire.

Dans ce chapitre, nous présenterons les systèmes de RV : leur définition, les logiciels à développer et les technologies d'interface. Ensuite, nous discuterons des applications de la technologie RV dans les domaines de l'éducation, des sciences et de l'ingénierie. L'impact de ces applications sera présenté avant la conclusion.



## 2.2. Systèmes de Réalité Virtuelle

Un système informatique qui implémente la technologie de RV, est considéré comme un système de RV. Selon [Youngblut 96], un système de RV est un monde généré par l'ordinateur, avec lequel l'utilisateur peut interagir de façon naturelle.

En général, un système de RV qui est aussi connu comme environnement de RV, se compose du monde virtuel et des outils pour visualiser ce monde virtuel et pour interagir avec lui. Le monde virtuel se construit à l'aide des logiciels de RV.

Fondamentalement, l'architecture d'un système de RV contient les quatre éléments suivants :

1. Un processeur d'entrée qui contrôle les outils utilisés pour donner les informations à l'ordinateur.
2. Un processeur de simulation qui s'occupe des interactions, des comportements des objets visualisés et de l'état du monde virtuel.
3. Un processeur d'interprétation qui crée la sensation de sortie à l'utilisateur.
4. Une base de données qui stocke les informations sur les aspects des objets visualisés et du monde virtuel.

Le temps de réponse de chaque élément principal a une influence sur le réalisme d'un événement dans le monde virtuel. Le cœur du système de RV est le processeur de simulation dont la réactivité doit rester compatible avec les événements du monde virtuel.

### 2.2.1. Les capacités des systèmes de Réalité Virtuelle

Un système de Réalité Virtuelle (ou la technologie de RV) devient un sujet très intéressant pour le développement d'un système de simulation. En effet, il est capable :

1. D'offrir une interface plus intuitive pour l'Interaction Homme-Machine.
2. De créer des applications qui ne sont pas réalisables dans le monde réel, soit parce qu'elles sont trop chères, soit parce qu'elles sont trop dangereuses.
3. D'être utilisé pour simuler, de façon intuitive, un événement qui ne peut pas être maîtrisé dans le monde réel, par exemple les événements intervenant à l'échelle atomique.

Grâce à la métaphore intuitive, l'utilisateur peut utiliser ses connaissances et ses compétences pour interagir avec le monde virtuel. En plus, l'expérience obtenue à partir d'un système de RV est directement transférable au monde réel. Un système de simulation développé en tant que système de RV permet de créer une simulation interactive aussi bien qu'un prototype interactif.

### 2.2.2. Types de systèmes de Réalité Virtuelle

Concernant la position de l'utilisateur par rapport au monde virtuel, les systèmes de RV se divisent en **systèmes de Réalité Virtuelle sans immersion** et en **systèmes de Réalité Virtuelle avec immersion**. Dans un système sans immersion, l'utilisateur regarde le monde virtuel de façon indirect via un écran et utilise un clavier ou une souris pour interagir. Par contre, dans le système avec immersion, l'utilisateur est vraiment immergé dans le monde virtuel.

**Les systèmes de réalité augmentée** appartiennent à un autre type de systèmes de RV. Ils superposent un objet virtuel ou un monde virtuel, au monde réel. L'objectif de ce système est de compléter le monde réel par des informations utiles qui normalement n'y sont pas disponibles.

**Les systèmes de réalité mixée** (mixed reality) sont des systèmes de RV où le monde virtuel est lié à un outil qui permet de manipuler les événements dans le monde réel présenté [Isdale 93]. Dans ce cas, il est donc possible d'avoir une interaction qui ne sera limitée, ni par le problème de la distance, ni par celui du temps.

### 2.2.3. Objets dans les systèmes de Réalité Virtuelle

Bien entendu, les objets dans le système de RV ne sont pas statiques. Ce sont des objets interactifs qui savent réagir lorsque l'utilisateur ou d'autres objets interagissent avec eux. Par conséquent, pour créer ces objets, on doit définir leurs comportements ainsi que leurs géométries.

Pour des raisons de réalisme, les objets sont évidemment modélisés par une géométrie identique à celle qu'ils possèderaient dans le monde réel. Leur positionnement dans le monde virtuel nécessite des informations sur leur position et leur orientation, et leur aspect, des informations sur leur couleur et leur texture.

## 2.3. Technologies d'interface de Réalité Virtuelle

Le type d'un système de RV et ses capacités dépendent des technologies d'interface de RV utilisées. Ces technologies comprennent les outils qui sont utilisés pour visualiser et interagir avec le monde virtuel.

A part l'interface visuelle, le son et l'odeur sont d'autres techniques augmentant le réalisme et confortant la sensation de présence de l'utilisateur dans le monde virtuel. Il existe deux techniques pour que l'utilisateur se sente vraiment immergé dans le monde virtuel. La première est le *Head Mounted Display*, ou casquette, qui contrôle le point de vue de l'utilisateur (figure II-1). L'autre, plus confortable, consiste à installer une salle possédant plusieurs grands écrans.

La première génération de cette deuxième technique est CAVE (*Cave Automatic Virtual Environment*) développée par *Electronic Visualization Laboratory* (EVL) de l' *University of Illinois* en 1992. En 1997 le *Japan Intelligent Modeling Laboratory* (IML) de l' *University of Tokyo* développe CABIN (*Computer Augmented Booth for Image Navigation*). A la différence de CAVE, CABIN possède un écran au plafond ainsi qu'un écran sur le sol, et le point de vue libre (*free of view point*) est de  $270^{\circ}$  (figure II-2). La troisième génération est COSMOS (*Cosmic Media of Six Screens*) développé par *VR Techno-Plaza, Gifu Prefecture* en 1998. Plus avancé que CABIN, COSMOS a six écrans et un point de vue libre de  $360^{\circ}$ .

En ce qui concerne les outils de l'interaction, cela inclus :

1. Les outils primaires (*Primary user input*) pour interagir avec le monde virtuel, tels que gant, *joystick*, *trackball* et souris 3D.
2. L'outil pour tracer le mouvement de l'utilisateur (*Tracking*), par exemple *eye tracking* (figure II-4)
3. L'outil qui fournit le tactile et le feed-back cinématique apparaissant lorsque l'utilisateur établit un contact avec les objets dans le monde virtuel (*Haptic*). Ce sont par exemple les dispositifs appelés *mechanical armature* et *optical position tracking*.
4. L'outil qui permet de voyager (*Full body motion*) dans le monde virtuel (dans un système de RV avec immersion). On distingue deux types de déplacements : le déplacement actif et le déplacement passif. Avec le premier type, l'utilisateur peut voyager librement, alors qu'avec le deuxième, il voyage dans son monde virtuel à l'aide d'un véhicule (figure II-5).

D'après [Youngblut 96], toutes les technologies actuelles d'interface de RV possèdent des restrictions, même celles incluses dans la catégorie *primary user input*. Aucune, parmi elles, n'a de rapport avec la modalité sensorielle de l'homme.



Figure II - 1 *Head Mounted Display*

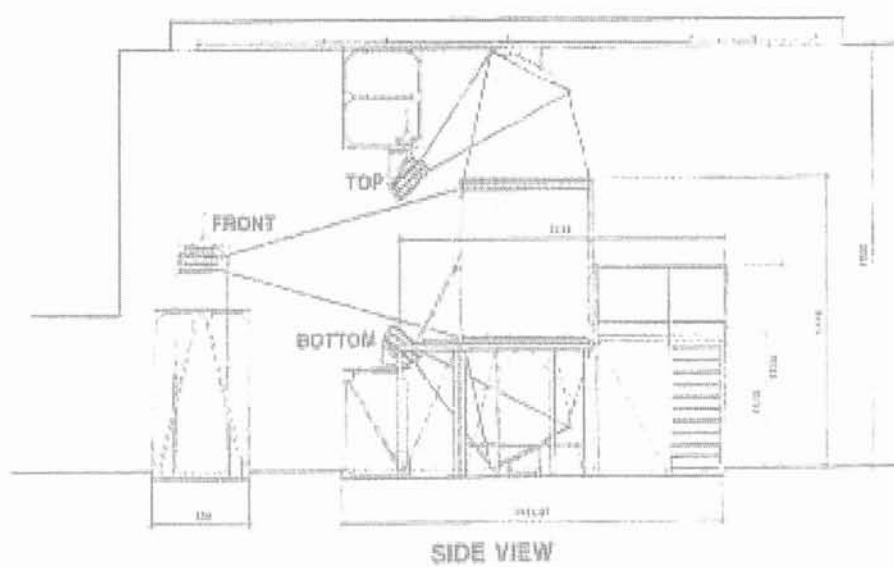


Figure II - 2 CABIN

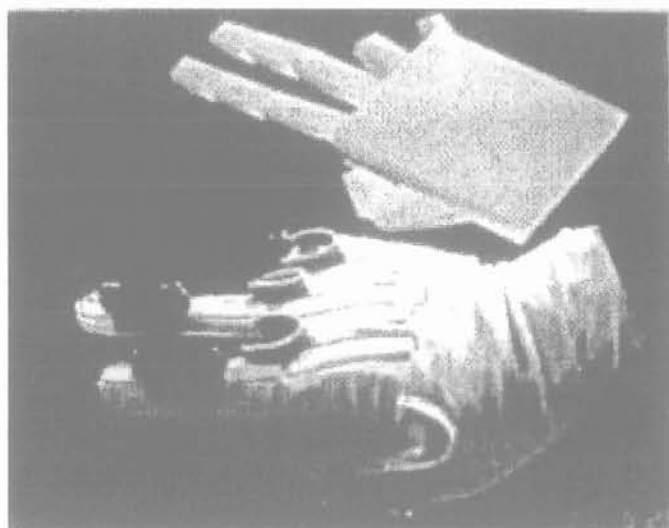


Figure II - 3 Gant

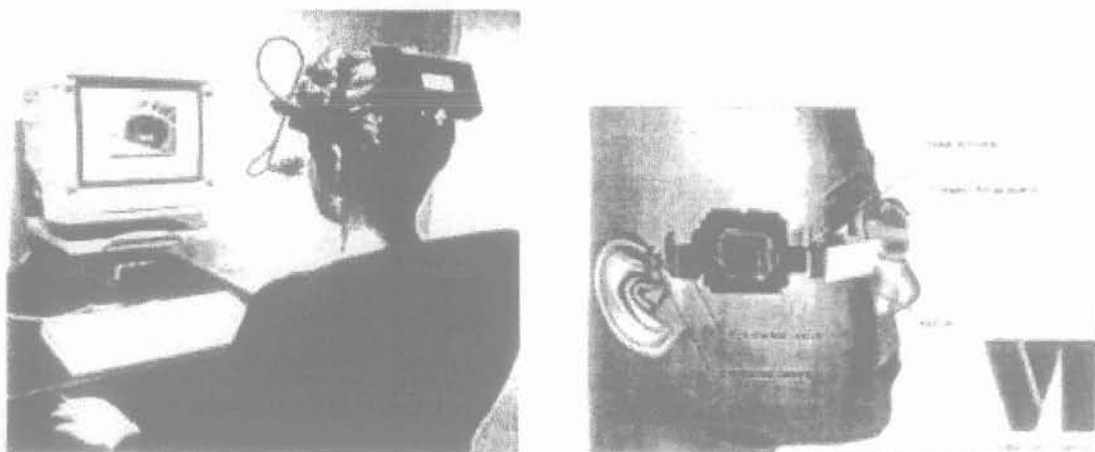


Figure II - 4 Eye tracking



Figure II-1 Full body motion

## 2.4. Logiciels de Réalité Virtuelle

Les logiciels de RV sont développés, soit à partir de bibliothèques de codes (*toolkits*) dans des langages tels que C++, C, Java3D et VRML, soit à partir de programmes déjà prêts (*authoring tool*) tels que VREAM et dVISE, qu'il convient de compléter avec l'interface graphique. A la différence des *authoring tools* dont l'objectif de développement de systèmes de RV est limité, les *toolkits* sont prévus pour le développement des systèmes de RV complexes. Nous présenterons une description brève de VRML et de Java3D, deux *toolkits* que nous avons choisis pour le développement de notre système, dans les sous chapitres 2.4.1 et 2.4.2.

### 2.4.1. VRML

VRML (*Virtual Reality Modelling Language*) est souvent mentionné comme une norme pour créer un objet 3D interactif. Cette norme a été élaborée en respectant le développement des applications sur l'Internet, l'Intranet et les systèmes *local-client* [VRML 97]. Les applications de VRML sont visualisées à l'aide d'un navigateur et d'un *browser* VRML. La norme VRML repose sur un principe de balises analogue à celui de la norme HTML (*Hyper Text Markup Language*) et supporte la transmission des informations complexes en 3D.

Nous avons utilisé ce langage de programmation dans notre première étude, en construisant une application prédéfinie pour le contrôle de vitesse d'un moteur asynchrone. Grâce à la structure de VRML, il est relativement facile de définir des objets possédant des géométries quelconques. Le seul inconvénient de ce langage est que l'on doit créer des scripts dans un langage de programmation plus généraliste (par exemple Java ou Java Script) pour le développement d'un système de RV complexe [Dewi 98].

### 2.4.2. Java3D

Java3D a été conçu pour apporter une indépendance par rapport à la technologie de RV. Les mondes virtuels pourront ainsi être visualisés en utilisant n'importe quel niveau de technologies d'interface de RV.

Java3D API est complémentaire de Java API. Java3D fournit les outils de base pour créer les mondes tridimensionnels. Il repose sur une programmation orientée objet de haut niveau. Il apporte un support pour la réutilisation des objets définis dans les systèmes de CAO ou les systèmes d'animation. Cependant, Java3D ne supporte pas directement tous les aspects possibles pour une application en 3D, mais les fonctions manquantes peuvent être obtenues par ajout de codes Java [Sowizral 97].

Pour les objets Lightwave 3D, la bibliothèque de Java3D inclut déjà le package pour le traitement des fichiers de format IFF. Par contre, pour atteindre les fichiers VRML, il faut compléter Java3D par un package externe actuellement disponible.

Parmi les points forts de ce logiciel, nous verrons que son modèle de programmation "*scene graph*" est particulièrement bien adapté à la visualisation des dispositifs électriques lorsqu'ils sont décrits par décomposition structurelle. Basé sur ce modèle de programmation, les objets d'un monde virtuel sont arrangés dans une structure arborescente. Ils peuvent être visualisés indépendamment l'un de l'autre.

## 2.5. Applications

La technologie de RV a été appliquée dans plusieurs domaines. Les militaires ont été particulièrement prolifiques à propos de simulateurs et de modélisations de systèmes d'armes. En médecine, cette technologie est utilisée pour créer les modèles 3D des organes pour l'assistance chirurgicale. L'éducation, les sciences et l'ingénierie sont également des domaines très actifs pour lesquels nous allons montrer quelques applications de la RV.

### 2.5.1. Technologie de Réalité Virtuelle en Education

L'étude faite par [Youngblut 98] de l' *Institute for Defense Analyses* montre que la technologie de RV peut, dans le monde de l'éducation, renforcer la démarche constructiviste. Selon cette méthode pédagogique, les étudiants peuvent mieux comprendre les concepts théoriques présentés dans la classe, s'ils ont la possibilité de réaliser des expériences.

Dans ce domaine, il existe deux types d'applications. Avec les applications du premier type, les étudiants ne peuvent réaliser qu'une expérience prédéfinie. Avec celles du deuxième type, ils peuvent eux-mêmes créer leur propre monde virtuel. Ce deuxième type est plutôt destiné aux activités de recherche.

L'évaluation de l'impact de ces applications conclut que les applications de RV peuvent améliorer les performances et la motivation des étudiants. Les applications développées en RV avec immersion sont plus pertinentes que celles développées en RV sans immersion. Cependant, **l'essentiel de l'application de la RV, dans le domaine de l'éducation, est dans l'interactivité** [Youngblut 98].

Concernant les utilisations pédagogiques courantes, celles sur ordinateur personnel (PC) sont de nos jours plus accessibles et donc plus faciles à mettre en œuvre que celles nécessitant une technologie d'immersion. Mais, un système de RV avec immersion est plus puissant pour établir des événements qui ne peuvent pas être définis ou qui sont difficiles à définir dans le monde réel. Soit parce qu'ils n'ont pas de forme physique, soit parce qu'ils ont une taille à l'échelle atomique ou planétaire, soit parce qu'ils ne peuvent pas être ressentis par l'homme.

### 2.5.2. Technologie de Réalité Virtuelle en Sciences

Dans le domaine des sciences, la technologie de RV est utilisée pour élaborer une visualisation scientifique. Pendant la phase initiale de recherche, le chercheur a besoin d'un outil qui permet d'interagir avec ses données et d'en mieux comprendre le contenu. Ensuite, vient la phase de description et de diffusion durant laquelle la visualisation scientifique peut être utilisée pour expliquer les idées et les concepts inventés. Cet effort de visualisation scientifique permet de toucher un public plus large qui sans cela se limiterait aux seuls spécialistes du sujet.

La visualisation scientifique offre la capacité d'explorer et de comprendre les informations et les données de manière graphique. Les possibilités de la technologie de RV sont considérables. L'interface d'un système de RV autorise une pleine immersion dans les données. En plus, cette technologie permet de créer une simulation interactive des phénomènes scientifiques.

Nous prenons *Virtual Physics Laboratory* développé par *Virtual Environment Technology Laboratory* (VETL) comme l'exemple de l'application de RV exprimant les concepts scientifiques. Cette application développée pour le scientifique est largement utilisée dans le domaine de l'éducation. L'objectif actuel de ce *Virtual Physics Laboratory* est de fournir une bonne interprétation des lois de Newton, de Maxwell et de Pauling. D'après [Bowen R], **l'habileté des étudiants dans la manipulation des formules exprimant les concepts scientifiques n'est pas significative de leur niveau de compréhension de ces concepts.**

Le monde virtuel de Newton dans le *Virtual Physics Laboratory* permet d'explorer les phénomènes physiques de la loi de Newton. Ce *Virtual Physics Laboratory* est un système de RV avec immersion utilisant une casquette et un gant. Dans ce monde virtuel, il est possible d'observer les déplacements aussi bien que la conservation de l'énergie cinétique et de la quantité de mouvement.

Les résultats de l'utilisation de *Virtual Physics Laboratory* montrent que les étudiants peuvent améliorer les connaissances initialement obtenues de manière classique [Dede CH].

### **2.5.3. Technologie de Réalité Virtuelle en Ingénierie**

Les applications de la RV dans le domaine de l'ingénierie sont très nombreuses. Elles incluent d'ailleurs certains aspects de l'éducation et des sciences. Nous allons décrire les applications de la technologie RV, et ses impacts dans ce domaine, en nous référant au processus d'engineering : le développement de produit, la gestion de production, la logistique, la maintenance et les utilisations des produits.

Dans le domaine de l'Ingénierie, l'application de la technologie de RV a pour objectif, en particulier, de résoudre les problèmes apparaissant pendant le processus de conception. L'élaboration du produit s'appuie alors sur des logiciels de conception capables de construire, soit une simulation interactive, soit un prototype interactif.

En ce qui concerne la gestion de production, la logistique, la maintenance et les utilisations du produit, cette technologie est appliquée, en particulier, pour résoudre la formation des personnes (*training*). Des systèmes de RV ont été développés permettant ainsi aux stagiaires, qui sont, soit les ingénieurs, soit les techniciens, soit les clients, d'étudier un produit ou un système de manière plus réaliste.



[Hadipriono 96] présente, par exemple, des applications pour la formation en génie Civil, où la technologie de RV permet de créer un système pour conduire une étude qui est chère et dangereuse à réaliser dans le monde réel.

## 2.6. Système de Réalité Virtuelle et CAO

Un système de RV possède la capacité de créer un modèle réaliste et géométriquement précis. La différence avec un système de CAO n'est plus facile à discerner.

La distinction est difficile en ce qui concerne la structure des données géométriques. Elle apparaît plus facilement si l'on s'intéresse aux finalités de chaque système [CSA 00]. Les modèles créés dans le système de RV sont uniques. La valeur ajoutée de ce système est dans la qualité de la visualisation des modèles.

En fait, un système de CAO et un système de RV sont des technologies complémentaires l'une de l'autre. Le premier est une technologie de création et de modification des données qui recherche une amélioration technique de leur visualisation. Au contraire, le deuxième est une technologie de la visualisation qui a besoin des données à visualiser. Le seul problème de cette utilisation complémentaire est que les fichiers du système de CAO ne sont pas toujours facilement lisibles par le système de RV !

Au-delà de la visualisation, un système de RV peut supporter aussi la création de modèles. Alors, la procédure classique de saisie des données peut être remplacée par le dessin et la manipulation d'objets dans le monde virtuel. Dans ce cas, le système de RV s'intègre totalement dans le système de CAO.

## 2.7. Visualisation interactive en électromagnétisme

Quelques études de visualisation interactive et intuitive en Génie Electrique ont déjà été menées. Parmi ces études, nous mentionnerons les recherches faites à l'Ecole Centrale de Lyon [Buret 96] [Thollon 99], à la faculté d'ingénierie de l'Université de Hiroshima - Japon [Okayama 99], à l'Université de la Technologie de Graz - Autriche [Preis 94], ... Les applications développées sont plus particulièrement dédiées à la visualisation des phénomènes électromagnétiques : distribution de champs magnétiques dans un circuit magnétique par rapport à la position de la bobine, à la géométrie de la tôle, aux caractéristiques du matériau magnétique, à l'entrefer, etc.... Ces outils de visualisation des résultats de simulations numériques, sont principalement destinées à mieux comprendre les phénomènes physiques de l'électromagnétisme. Leur utilisation est essentiellement pédagogique.

Une interface graphique classique, la technologie de RV et des techniques particulières de réduction des temps de calcul sont implantées afin de pouvoir interpréter correctement et interagir en temps réel, avec les résultats de simulation.

## 2.8. Conclusion

Les applications développées dans de nombreux domaines prouvent que la technologie de RV permet d'avoir une visualisation plus interactive, plus intuitive et plus compréhensible que les technologies statiques usuelles.

Nous sommes intéressés par l'application de cette technologie en Génie Electrique pour expliquer les phénomènes qui expriment le fonctionnement d'un dispositif électrique. Les systèmes de simulation de notre discipline concernés devraient être complétés par des interfaces de RV permettant d'observer l'intérieur des dispositifs étudiés aussi bien que leurs performances.

La réalisation d'un tel système nécessite évidemment des efforts importants et ne peut se faire que progressivement. Dans une première étape, nous prévoyons d'utiliser cette technologie en développant un système de RV sans immersion. Nous utiliserons les interfaces de RV simples : un écran ordinaire, une souris et un clavier. Bien entendu, avec cette interface nous n'obtiendrons qu'une interaction réduite au minimum, mais le but principal de notre démarche initiale est d'examiner les concepts que l'on doit manipuler dans un système de RV dédié au Génie Electrique. En particulier, notre système doit être développé en considérant qu'il doit s'intégrer, à terme, dans un système de CAO en Génie Electrique.

Nous utilisons Java3D pour construire le monde virtuel de notre système, car cet outil se veut indépendante de la technologie RV. Nous espérons que les investissements initiaux réalisés dans une technologie RV minimaliste pourront être réutilisés dans le cadre de technologies plus évoluées.

Concernant les objets virtuels de notre système de RV, ce seront des dispositifs électriques dont nous modéliserons la géométrie et les propriétés physiques ainsi que les comportements physiques ou mécaniques. Les candidats sont donc les dispositifs classiques de l'électrotechnique : contacteurs électromécaniques, machines électriques tournantes, transformateurs électriques ou bien les dispositifs dans lesquels le champ électromagnétique joue un rôle fondamental : têtes d'enregistrement magnétique, micro actionneurs, ... Les opérations que nous envisageons de réaliser permettront d'analyser le comportement global du dispositif : évolution de la force de rappel, du couple électromécanique, ... ou bien les phénomènes électromagnétiques locaux : trajet et intensité du champ magnétique, état de la saturation magnétique, densité de pertes, ...

Toutes ces analyses sont dès aujourd'hui accessibles grâce aux outils de simulation actuels. Notre intention est d'apporter un point de vue différent par rapport à la démarche d'analyse traditionnelle. En effet, les outils de simulations traditionnels sont orientés « méthode de calcul ». Autrement dit, ils permettent la simulation d'un dispositif quelconque, grâce à une implantation efficace d'une ou de plusieurs méthodes numériques. Notre objectif est de promouvoir une approche orientée métier dans laquelle le centre est, non plus le calcul numérique, mais le dispositif étudié et les pratiques et contraintes attachées à ce dispositif.



## Chapitre 3

# **Modélisation du Laboratoire d'Expérimentation en Electrotechnique**



## Chapitre 3

### Modélisation du Laboratoire d'Expérimentation en Electrotechnique

#### 3.1. Définition

L'expérimentation est une observation contrôlée [Oren 78]. C'est ainsi que l'on procède sur des dispositifs électriques dans le Laboratoire d'Expérimentation en Electrotechnique. Le but des expérimentations est, soit d'étudier, soit d'analyser le fonctionnement des dispositifs électriques par leurs comportements physiques. Dans son organisation, le Laboratoire d'Expérimentation en Electrotechnique est une salle contenant tout le matériel nécessaire pour mettre en œuvre des manipulations d'expérimentation. Il s'agit des dispositifs électriques, des instruments de mesure, des sources électriques et des outils de connexion. Pour l'expérimentation normalisée, une manipulation est normalement complétée par une norme d'expérimentation (standard d'expérimentation).

Le processus d'expérimentation est constitué, en principe, des étapes suivantes (figure III - 1) :

- *Définition du problème;*  
Une expérimentation commence par le choix du dispositif examiné et de ses comportements physiques à observer.
- *Développement de la manipulation d'expérimentation;*  
Après la définition du problème, l'étape suivante est le développement de la **manipulation d'expérimentation** qui est l'assemblage du dispositif électrique, de ses attributs (charge, auxiliaires, ..), des sources électriques et des instruments de mesure.
- *Mise en œuvre de l'expérimentation;*  
L'expérimentation est mise en œuvre en alimentant le dispositif en courant ou en tension électrique.
- *Observation;*  
Dès que le dispositif fonctionne, l'étape suivante est l'observation de ses comportements physiques par des réactions physiques visibles du dispositif en lui-même et des instruments de mesure.
- *Analyse;*  
À partir des résultats obtenus durant l'observation, l'analyse permet de vérifier leur adéquation par rapport à la synthèse théorique ou les spécifications que devrait remplir le dispositif.

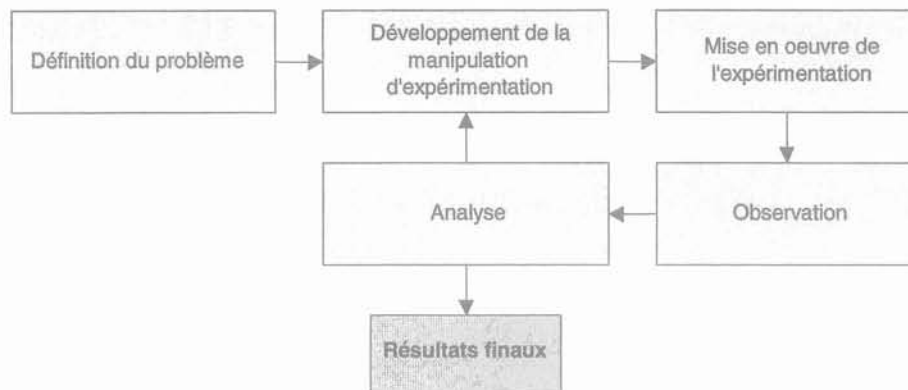


Figure III - 1 Processus d'expérimentation

Concernant la procédure, nous constatons qu'il existe deux types d'expérimentation: ***l'expérimentation normalisée*** et ***l'expérimentation non normalisée***. Dans le cas de l'expérimentation normalisée, on connaît déjà les paramètres influents pour certaines performances du dispositif examiné et les étapes particulières pour pouvoir l'observer. Ce type d'expérimentation est, d'habitude, effectué pendant les travaux pratiques par les étudiants ou, en milieu industriel, pendant les essais de réception, les essais de qualification ou d'acceptation, et les essais de certification [Picard 96]. Par contre, l'expérimentation non normalisée concerne l'expérimentation des nouveaux dispositifs électriques. Pour ceux-ci, les paramètres qui devraient être modifiés et les traitements particuliers pour observer certaines performances ne sont pas encore normalisés. L'expérimentation faite ne se base alors que sur le savoir-faire de l'expérimentateur. Par la suite, en validant les résultats obtenus et en rédigeant la documentation, cette expérimentation deviendra une expérimentation normalisée.

En définitive, la technique d'expérimentation dans un laboratoire est une méthodologie naturelle qui favorise l'analyse intuitive de l'homme lorsqu'il s'agit d'étudier les comportements physiques d'un objet. Nous avons proposé une transposition de cette technique, en particulier concernant l'expérimentation des dispositifs électriques, comme une méthode d'exploration pour les logiciels de CAO en Génie Electrique. Ceci devrait faciliter l'utilisation de ces logiciels qui deviennent de plus en plus complexes. Par ailleurs, cette approche peut aider à la transmission des travaux faits sur les logiciels, vers l'expérimentation dans un vrai laboratoire. Ainsi, les activités de modélisation, de simulation et d'analyse effectuées avec les logiciels de CAO sont assimilées au processus d'expérimentation. Nous concrétiserons cette approche en développant un système interactif qui représente le modèle du Laboratoire d'Expérimentation en Electrotechnique.

### 3.2. Difficultés

Il n'existe pas de formalisme bien défini à propos du Laboratoire d'Expérimentation en Electrotechnique. Les points fondamentaux et les points de détails qui doivent être définis, pour que son

modèle puisse être vraiment représenté, ne sont, ni explicités, ni formalisés. C'est pourquoi, nous avons développé une méthodologie nous conduisant à formaliser les composants, leurs caractéristiques, les relations entre ces composants et, en particulier, les connaissances impliquées dans une expérimentation de dispositif électrique (§ 3.3).

### 3.3. Méthodologie

La figure III-2 montre schématiquement la méthodologie que nous mettons en œuvre pour modéliser le Laboratoire d'Expérimentation en Electrotechnique. Afin de limiter le domaine nous formalisons d'abord la définition et les types d'expérimentations de dispositifs électriques. Une expérimentation est définie comme l'observation contrôlée des comportements physiques d'un dispositif électrique, en mettant en place ce dispositif dans les conditions adéquates. Comme nous l'avons indiqué précédemment, il existe deux types d'expérimentations : l'expérimentation normalisée et l'expérimentation non normalisée. Par la suite, nous nous plaçons dans le cadre de **l'expérimentation normalisée**. Ce choix est déterminé en considérant que la conception, à l'aide des logiciels de CAO, concerne l'amélioration des dispositifs existants, en modifiant leurs dimensions et leurs propriétés physiques (§ 1.3.1).

L'étape suivante consiste à formaliser les entités intervenant au cours d'une expérimentation de dispositif électrique. En principe, ce sont la manipulation d'expérimentation, la norme d'expérimentation et l'expérimentateur. En identifiant les composants de chaque entité, nous arrivons à la conclusion que la manipulation d'expérimentation elle-même est constituée du dispositif électrique, de ses attributs, des sources électriques et des instruments de mesure. Ils sont reliés suivant un montage particulier. Ensuite, **la norme d'expérimentation** se compose des hypothèses, des contraintes et du protocole d'expérimentation. Les **hypothèses** contiennent les principes théoriques et les objectifs correspondant au fonctionnement d'un dispositif électrique. Les **contraintes** sont les conditions limites que devrait supporter un dispositif électrique. Ces conditions limites sont normalement déterminées pendant sa conception. Enfin, le **protocole d'expérimentation** est l'enchaînement particulier des opérations qui doivent être effectuées pour que l'on puisse observer certains comportements physiques du dispositif électrique. Cette analyse, nous a permis de déterminer les composants fondamentaux qui devraient être disponibles dans le laboratoire.

Les dispositifs examinés sont classifiés en trois catégories qui reflètent leur principe de fonctionnement : les appareillages, les transformateurs et les machines tournantes. Chaque catégorie est, ensuite, divisée en sous catégories. L'annexe B présente, en détail, les catégories et les sous catégories des dispositifs électriques. La structure de la norme d'expérimentation pour chaque objectif d'expérimentation sera présentée dans le sous chapitre 3.3.3.

Après la formalisation des entités, nous passons aux étapes principales d'une expérimentation dans un laboratoire. Ces étapes, regroupées dans le processus d'expérimentation, sont la définition des problèmes, le développement de la manipulation d'expérimentation, la mise en œuvre de



l'expérimentation, l'observation et l'analyse. Ensuite, nous distinguons les activités contenues dans chaque étape (tableau III -1). L'objectif d'expérimentation, qui correspond aux comportements physiques concernés à observer, est lié avec un **montage d'expérimentation**. Il s'agit, par rapport à la manipulation d'expérimentation, d'une certaine combinaison des sources électriques, des instruments de mesure et des attributs auxiliaires. Les objectifs d'expérimentation pour chaque catégorie de dispositifs seront décrits dans le sous chapitre 3.3.2.

La dernière étape de la modélisation du Laboratoire d'Expérimentation en Electrotechnique est de formaliser les connaissances intervenant lors d'une expérimentation. Par analogie avec les connaissances pour la conception [Gentil 88], cette connaissance peut être divisée en trois catégories : la connaissance descriptive, la connaissance déductive et la connaissance stratégique. La connaissance descriptive correspond aux informations relatives aux paramètres du dispositif examiné, introduites lors de la définition des problèmes. Dans ce cas, les dispositifs électriques sont considérés comme les problèmes qui doivent être résolus. Ensuite, la connaissance déductive contient les informations à propos de la règle d'expérimentation, c'est-à-dire la norme d'expérimentation. Enfin, la connaissance stratégique représente le savoir-faire de l'expérimentateur pour contrôler les étapes de l'expérimentation. Cette connaissance englobe donc le processus d'expérimentation et le protocole d'expérimentation qui sont prédéfinis dans le cas des expérimentations normalisées. Par contre, pour les expérimentations non formalisées, le protocole d'expérimentation, en particulier, n'est pas prédéfini. En plus du savoir-faire de l'expérimentateur, il contient également les connaissances concernant les comportements physiques du dispositif électrique.

En résumé, l'annexe A exprime, en tableau, les étapes de cette méthodologie et les résultats attendus à chaque étape.

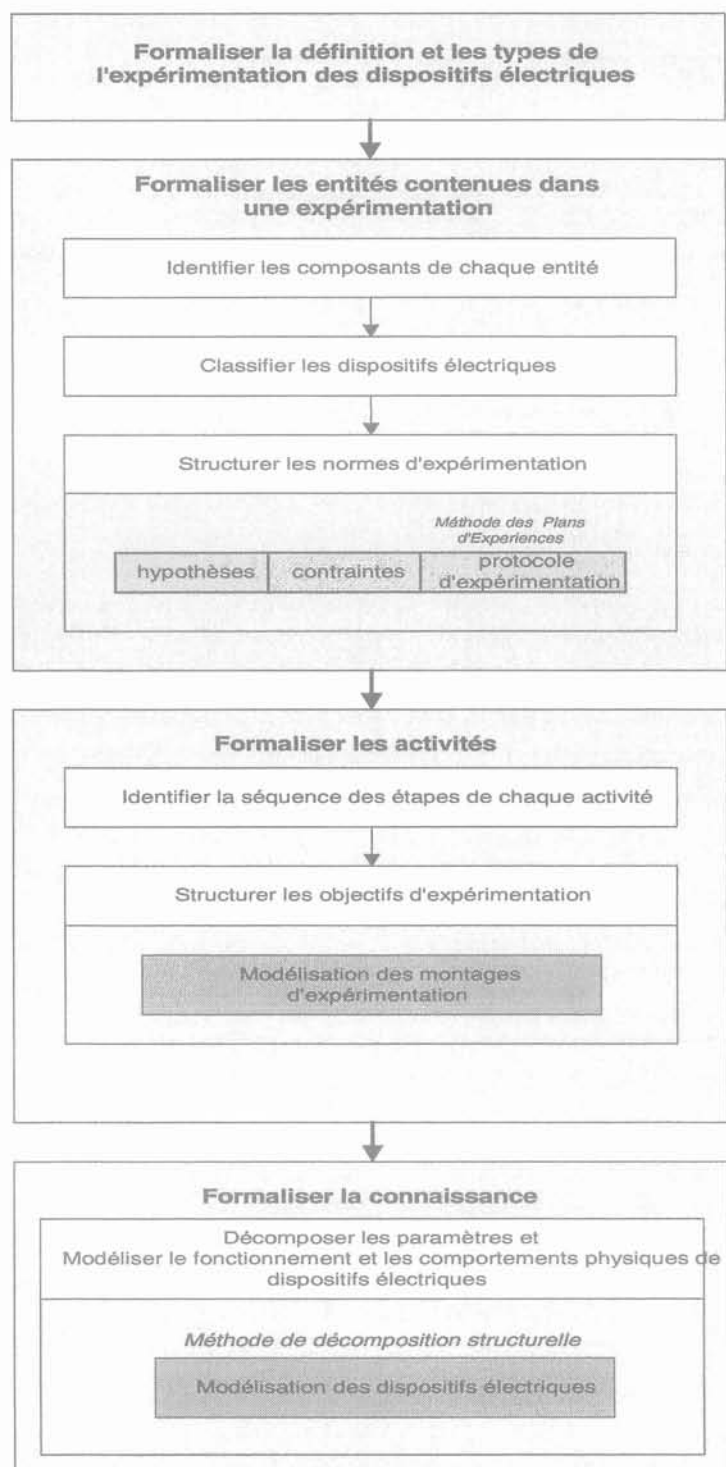


Figure III - 2 Méthodologie de modélisation du Laboratoire d'Expérimentation en Electrotechnique

Tableau III - 1 La séquence des activités de chaque étape du processus d'expérimentation

No.	Processus d'expérimentation	
	Etapes	Activités
1.	<i>Définition du problème</i>	<ul style="list-style-type: none"> <li>- Déterminer le dispositif électrique examiné.</li> <li>- Déterminer l'objectif ou les objectifs d'expérimentation.</li> </ul>
2.	<i>Développement de la manipulation d'expérimentation</i>	<ul style="list-style-type: none"> <li>- Construire l'assemblage du dispositif et de ses attributs, des sources électriques et des instruments de mesure.</li> <li>- Mettre en place les conditions opérationnelles du dispositif correspondant à certains objectifs d'expérimentation.</li> </ul>
3.	<i>Mise en œuvre de l'expérimentation</i>	<ul style="list-style-type: none"> <li>- Déterminer les composants du dispositif concernés.</li> <li>- Noter les caractéristiques de fonctionnement (les informations de la plaque signalétique), la géométrie du dispositif électrique.</li> <li>- Faire fonctionner le dispositif électrique.</li> <li>- Modifier les paramètres influents pour les comportements physiques observés</li> </ul>
4.	<i>Observation</i>	<ul style="list-style-type: none"> <li>- Observer les comportements physiques du dispositif, soit via des instruments de mesure, soit via ses réactions physiques.</li> </ul>
5.	<i>Analyse</i>	<ul style="list-style-type: none"> <li>- Evaluer les résultats obtenus par l'observation.</li> <li>- Exprimer les résultats en utilisant les théories concernées.</li> <li>- Tirer les conclusions.</li> </ul>

### 3.3.1. Modélisation de dispositifs électriques

Dans le Laboratoire d'Expérimentation en Electrotechnique, le dispositif électrique est, comme nous l'avons décrit précédemment, l'objet du problème qui doit être défini au début d'une expérimentation. Le modèle d'un dispositif électrique et la formalisation de ses paramètres sont développés en utilisant les concepts de décomposition de problème [Simon 68]. Il existe trois types de décompositions possibles pour un dispositif électrique [Gentil 88] :

- *Décomposition fonctionnelle;*

Le dispositif électrique est décomposé en sous composants selon ses fonctionnalités. Chacun de ces sous composants représente une des fonctionnalités du dispositif.

- *Décomposition structurelle;*

Le dispositif électrique est découpé en sous composants élémentaires selon sa structure physique. Cette décomposition conduit à une structure hiérarchique qui présente la relation entre un composant et ses sous composants.

- *Décomposition hybride;*

Il s'agit d'une combinaison des deux décompositions précédentes. Elle forme l'association entre des fonctionnalités et des structures physiques.

En admettant que la référence sur des structures physiques est la technique naturelle pour les scientifiques lorsqu'ils examinent un objet, nous avons décidé d'implanter la décomposition structurelle, afin de pouvoir favoriser l'analyse intuitive de l'expérimentateur. De plus, cette décomposition permet de structurer de façon hiérarchique, les paramètres d'un dispositif électrique et, par conséquent, de connaître les influences de chaque sous composant sur son fonctionnement.

Le fonctionnement d'un dispositif électrique est, en effet, affecté par la géométrie et les propriétés physiques de ses composants. Les paramètres d'un dispositif sont structurés en considérant qu'ils peuvent contenir des données nécessaires pour étudier tous ses domaines physiques possibles, soit expérimentalement, soit par simulation. Chaque modèle de composants de dispositif électrique possède donc la connaissance descriptive de sa géométrie et de ses propriétés physiques. D'une manière naturelle, au sommet de cette hiérarchie, la connaissance descriptive du dispositif est attachée à son modèle qui est constitué de tous les modèles de ses composants.

La visualisation naturelle souhaitée, a besoin de la modélisation des comportements physiques du dispositif électrique autant que de sa géométrie. Autrement dit, le modèle choisi doit vraiment représenter les performances physiques comme dans la réalité. Comme dans le laboratoire réel, il est alors possible d'observer les comportements physiques du dispositif qui peuvent varier en fonction du temps.

Les comportements physiques d'un dispositif électrique sont modélisés en examinant ses réactions dans chacun des états possibles au cours de son fonctionnement. La séquence des événements est importante ainsi que la connaissance précise des états atteints. Les dispositifs que nous avons choisi d'étudier, les appareillages, les transformateurs et les machines tournantes, ont pour objectif principal la transformation de l'énergie électrique en énergie électrique, de l'énergie électrique en énergie mécanique ou l'inverse. L'état de l'un de ces dispositifs électriques peut donc être défini dans les deux domaines, électrique et mécanique. Du point de vue électrique, le dispositif peut se situer, soit dans l'état transitoire électrique, soit dans l'état permanent électrique. De même, il peut se placer soit dans l'état transitoire mécanique, soit dans l'état permanent mécanique.

L'état transitoire électrique peut se produire lorsque le dispositif est alimenté progressivement. C'est le cas lorsque le composant conducteur du dispositif a besoin de temps pour atteindre une certaine valeur de courant. Il peut également être la conséquence d'un état transitoire mécanique qui influe sur les paramètres électriques. Par exemple, le déplacement de l'armature du contacteur entraîne la variation de son inductance et, par conséquent, la variation du courant d'alimentation. Au contraire, le dispositif est considéré dans l'état permanent électrique lorsque le courant ou la tension d'alimentation ne change pas en fonction du temps. Dans le cas d'une alimentation périodique, on considère que le dispositif a atteint son régime permanent lorsque chaque nouvelle période est semblable à la précédente.

Lorsqu'un composant du dispositif se déplace en translation, en rotation ou suivant un mouvement quelconque avec une vitesse variant dans le temps, il est dans un état transitoire mécanique. L'état permanent mécanique est atteint lorsque aucun composant ne bouge ou lorsque toutes les vitesses sont constantes.

Par ailleurs, le type de l'alimentation, c'est-à-dire courant continu ou courant alternatif, et les caractéristiques des matériaux magnétiques déterminent les comportements physiques d'un dispositif électrique. Dans les sous chapitres suivants (§ 3.3.1.1 et § 3.3.1.2), nous décrirons, à titre d'exemple, la modélisation d'un contacteur électromécanique.

#### *3.3.1.1. Un exemple : Les composants du contacteur électromécanique*

La figure III-3 présente la structure des composants d'un contacteur électromécanique. En principe, un contacteur se compose de l'additif, de l'encombrement (couverture), de l'électroaimant et des pôles. L'électroaimant lui-même est découpé en circuit magnétique, bobine, spires de Frager, pôles et ressort de rappel. De même, les pôles sont subdivisés en contacts et ressorts des pôles. Le modèle du contacteur est donc formé hiérarchiquement à partir de modèles de ses sous composants élémentaires.

Grâce à cette structure, on peut accéder aux informations d'une structure de bas niveau, à partir d'une structure de plus haut niveau. Ainsi, il est facile de comprendre ce que sont les données contenues dans un contacteur électromécanique. En outre, cette structure permet de connaître intuitivement le rôle de chaque composant du contacteur sur son fonctionnement. Cette interaction intuitive, entre rôle et connaissance descriptive, peut être avantageusement exploitée dans un système de Réalité Virtuelle.

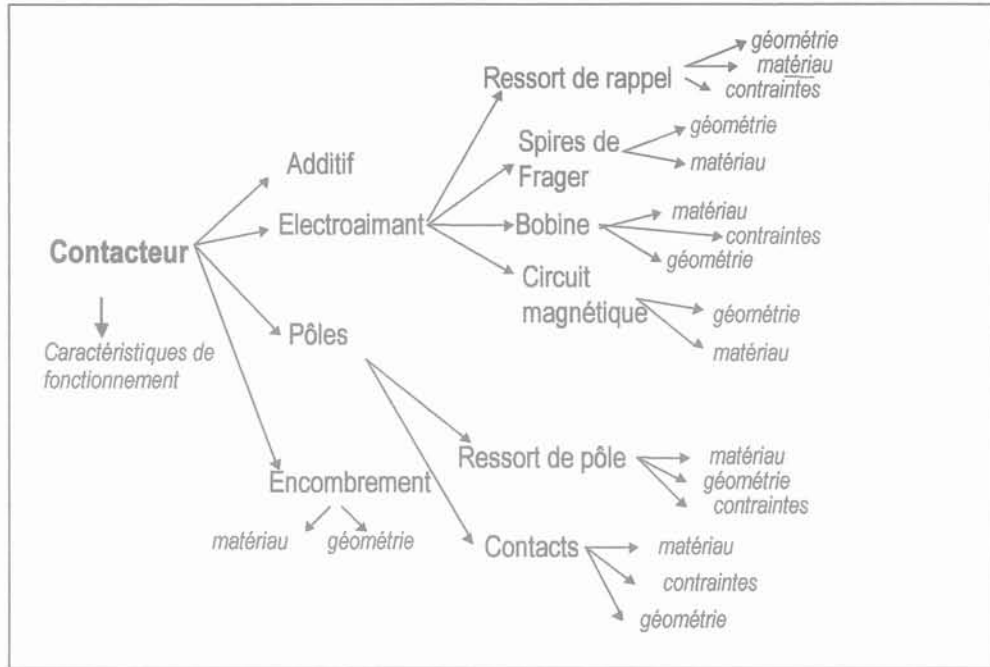


Figure III - 3 Décomposition structurelle du contacteur électromécanique

### 3.3.1.2. Un exemple : Les comportements physiques du contacteur électromécanique

Les comportements physiques d'un dispositif électrique sont les événements émis pendant son fonctionnement. Ils s'inscrivent pour notre cas dans les domaines électromagnétique, mécanique et thermique (§ 1.2). Pour modéliser les comportements physiques d'un contacteur électromécanique, nous partons de ses états mécaniques. Ensuite, dans chaque état, ils sont classifiés en considérant le type d'alimentation et le caractère des matériaux magnétiques. L'annexe C montre, chronologiquement, ces comportements physiques. Leurs modèles mathématiques et les hypothèses correspondantes sont présentés dans l'annexe D.

Le contacteur électromécanique réagit différemment suivant qu'il est alimenté en courant continu ou en courant alternatif, comme cela est représenté sur la figure III-4 [Féchat 95]. Le comportement mécanique représente la performance finale. Cependant, il influence le comportement électrique, en particulier pour l'alimentation en courant continu. Par ailleurs, le comportement thermique qui dépend des caractéristiques des matériaux, est issu du comportement électrique.

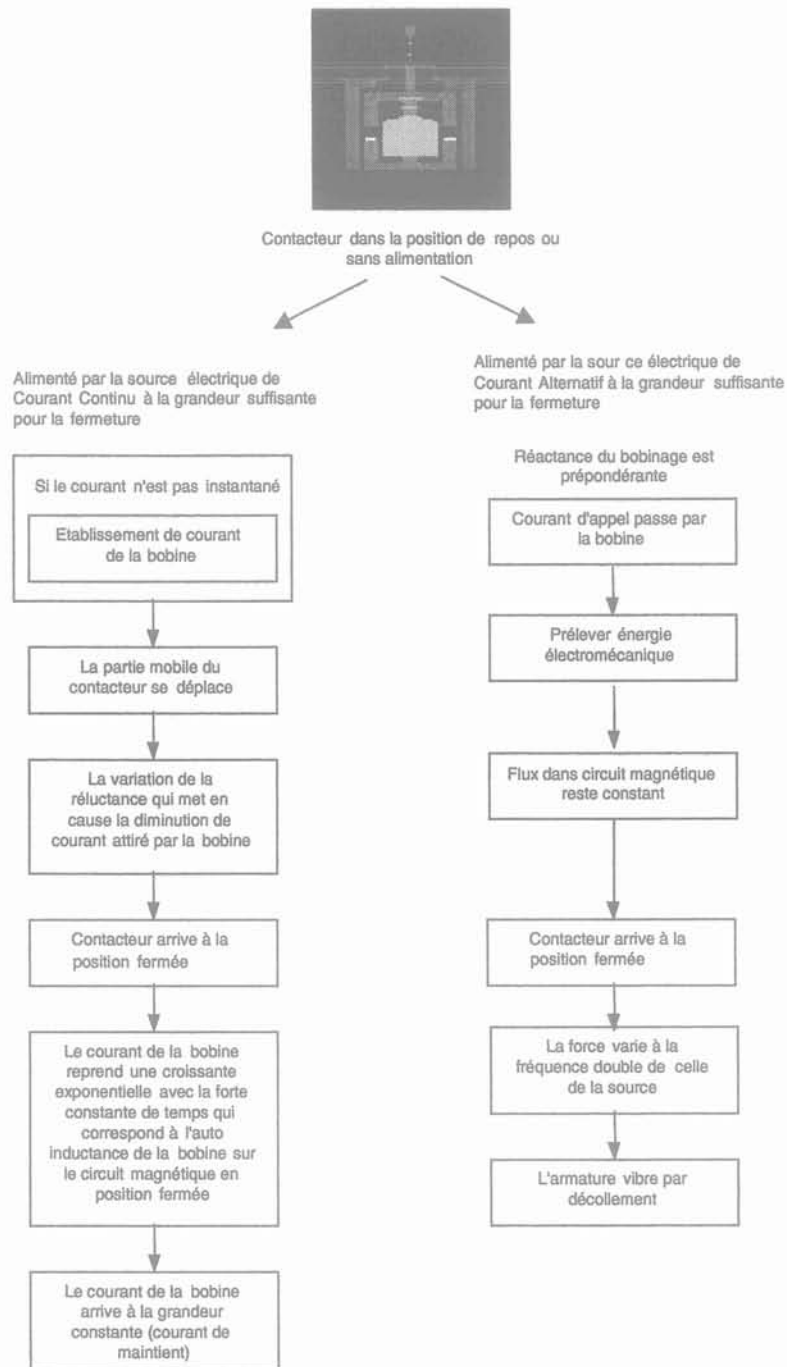


Figure III - 4 Schéma de fonctionnement d'un contacteur électromécanique

Parmi les comportements physiques, il en existe qui ne peuvent pas être visualisés sans l'aide d'instruments de mesure. Il s'agit des phénomènes électromagnétiques, comme par exemple le champ magnétique, les pertes magnétiques, les courants de Foucault, ... Les autres, tels que l'arc électrique, le mouvement de l'armature, l'augmentation de température, ..., peuvent être observés en regardant, en touchant, et en écoutant le contacteur. En implantant la technologie de Réalité Virtuelle, il est possible

d'observer ces comportements physiques de manière naturelle. De plus, cette technologie nous permet de sentir les comportements physiques qui sont, dans le monde réel, uniquement accessibles par l'intermédiaire des instruments de mesure. Ainsi, à l'aide de l'interface de Réalité Virtuelle (§ 2.3), on pourra regarder ou sentir l'existence du champ magnétique à l'intérieur des dispositifs électriques.

En résumé, en plus de la connaissance descriptive sur sa géométrie, ses propriétés physiques et ses caractéristiques de fonctionnement, un modèle de contacteur doit contenir aussi la connaissance stratégique : il doit connaître sa façon de réagir lorsque l'expérimentateur lui fait subir un traitement.

### **3.3.2. Formalisme des objectifs d'expérimentation**

Le formalisme des objectifs d'expérimentation d'un dispositif électrique peut être vu comme le classement d'expérimentation en considérant la nature physique du dispositif électrique [Picard 96]. Pour un dispositif électrique, les objectifs d'expérimentation explicitent les domaines physiques que l'on veut examiner. Les objectifs d'expérimentation sont classifiés en trois catégories suivant le classement des dispositifs électriques : les objectifs d'expérimentation de l'appareillage, ceux du transformateur et ceux de la machine tournante.

Les types des objectifs d'expérimentation sont choisis en examinant les facteurs influents lors de la conception de chaque catégorie de dispositifs électriques. Ensuite, nous déterminons les comportements physiques qui correspondent à un objectif d'expérimentation en distinguant lesquels sont les réponses principales de cet objectif et lesquels sont les réponses additionnelles. Au-delà, nous pouvons déduire des paramètres d'entrée, des conditions d'expérimentation, des hypothèses, des contraintes et des montages d'expérimentation (§ annexe E). Dans le cas des expérimentations normalisées, ces informations sont obtenues grâce aux références disponibles.

**Les paramètres d'entrée** sont les paramètres fondamentaux qui devront être modifiés durant l'expérimentation car ils interviennent sur le comportement physique concerné. **Les conditions d'expérimentation**, qui prennent part au protocole d'expérimentation, expliquent l'état du dispositif électrique dans lequel le comportement physique visé peut se produire.

### **3.3.3. Formalisme des normes d'expérimentation**

La norme d'expérimentation est, en fait, le texte ou la documentation régissant l'expérimentation. Dans l'industrie, pour ce qui concernent la réception, la qualification ou l'acceptation et la certification des produits, la création des normes d'expérimentation implique, souvent, les organismes de normalisation, soit nationaux soit internationaux comme par exemple la CEI (*Commission Electrotechnique Internationale*) et le CENELEC (*Comité Européen de Normalisation Electrotechnique*) [Picard 96]. Les normes d'expérimentation peuvent être considérées comme les manuels de laboratoire rédigés par les experts des expérimentations : soit les techniciens, soit les ingénieurs, soit les chercheurs.



Un objectif d'expérimentation s'explique par une norme d'expérimentation particulière. Elle réunit les paramètres d'entrée et les conditions d'expérimentation, les hypothèses et les contraintes obtenus lors de la formalisation des objectifs d'expérimentation (§ 3.3.2).

### **3.3.4. Formalisme des protocoles d'expérimentation**

Le protocole d'expérimentation englobe les paramètres d'entrée et les conditions d'expérimentation. Il explique la procédure particulière pour accomplir ces conditions d'expérimentation afin de pouvoir observer certains comportements physiques.

A propos des expérimentations non formalisées, nous remarquons que le protocole d'expérimentation peut être établi en utilisant les concepts de la méthode des plans d'expériences [Benoist 94] [Deonsant 96]. La méthode des plans d'expériences est un outil d'aide à l'expérimentation d'un problème du type boîte noire. Cette méthode est conçue pour toutes sortes de problèmes techniques : l'optimisation de la conception d'un produit, le réglage d'un moyen de production ou d'un processus, les investigations pour mieux comprendre un phénomène, et la mise au point d'une procédure d'essais. Elle a comme objectif de quantifier avec le maximum de précision, mais en réalisant le moins d'essais possibles, l'effet de plusieurs facteurs sur une ou plusieurs réponse(s).

Dans notre cas nous assimilons le nouveau dispositif examiné à un problème de type boîte noire. Nous considérons les étapes principales représentées sur la figure III-5. D'abord on formalise le problème. Ensuite, on détermine les grandeurs des réponses souhaitées et les facteurs influents relatifs. Pour nous, les réponses souhaitées sont exprimées comme les comportements physiques à observer. L'étape suivante est de construire le plan d'essais, c'est-à-dire les configurations des facteurs influents qui seront expérimentées. Ensuite, on utilise ces configurations pour effectuer les essais ou l'expérimentation et on observe les comportements du dispositif pour chaque configuration. Les résultats de cette expérimentation sont analysés pour quantifier les influences des facteurs et expliquer le fonctionnement du nouveau dispositif électrique, qui pour nous est une boîte noire. Enfin la configuration qui résout le problème nous donne les paramètres d'entrée et les conditions d'expérimentation. Les paramètres d'entrées correspondent aux modalités quantitatives des facteurs influents, alors que les conditions d'expérimentation en sont les modalités qualitatives. La procédure particulière d'expérimentation peut être déduite du fonctionnement du nouveau dispositif électrique, ce qui est en accord avec la configuration concernée.

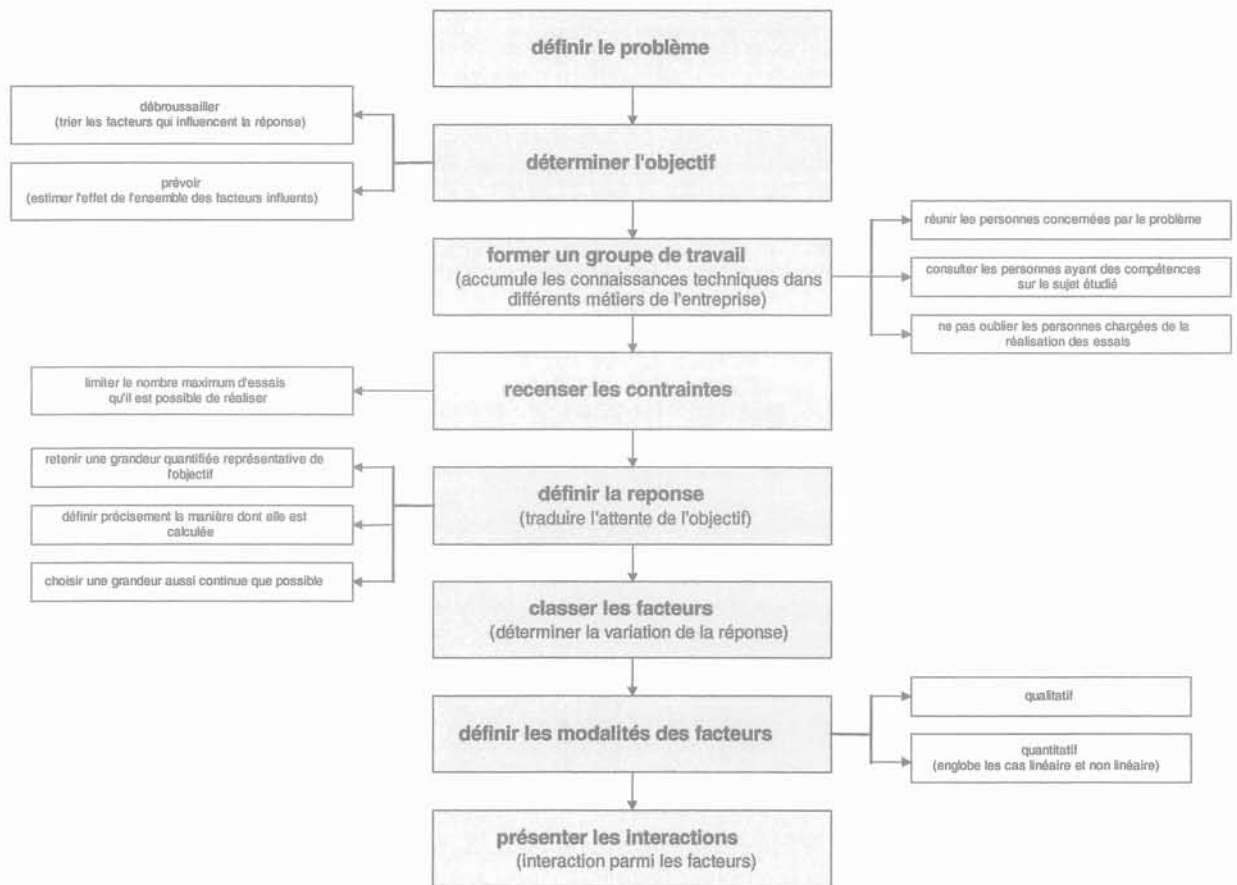


Figure III - 5 Formalisation du problème par la méthode des plans d'expériences.

### 3.4. Conclusion

Nous constatons qu'il existe trois ensembles fondamentaux qui doivent être formalisés lors de la modélisation du Laboratoire d'Expérimentation en Électrotechnique :

- les entités prenant part à une expérimentation,
- les activités d'une expérimentation,
- les connaissances impliquées dans une expérimentation.

Lors de ces formalisations, tout en considérant le caractère normalisé ou non normalisé de l'expérimentation de dispositif électrique, nous devons déterminer les points de détails importants pour que le modèle puisse être vraiment représentatif.

Si nous faisons un parallèle avec les étapes principales de la simulation (§ 1.2.3), la définition des problèmes et le développement de la manipulation d'expérimentation correspondent à la description du problème. Ensuite, la mise en œuvre de l'expérimentation s'accorde à la mise en œuvre de la méthode numérique. Enfin, l'observation et l'analyse représentent la dernière étape de la simulation, qui comprend

la vérification, la visualisation et l'interprétation des résultats de la simulation numérique. Le modèle du dispositif examiné définit des paramètres d'entrée nécessaire à la simulation. Le domaine physique à étudier est exprimé par l'objectif d'expérimentation. La technique d'expérimentation peut ainsi être adoptée comme un standard du métier pour naviguer dans les logiciels de CAO en Génie Electrique. De plus, elle permet de définir une structure d'informations grâce à laquelle il est facile de comprendre ce que sont les données contenues dans le dispositif et, par conséquent intuitivement, de connaître leurs rôles dans le fonctionnement du dispositif.

A la conception de nouveaux dispositifs électriques, il s'agit soit de modifier les dimensions ou les propriétés physiques d'un dispositif existant, soit de concevoir vraiment un nouveau dispositif dont aucune caractéristique géométrique, physique ou de fonctionnement n'est disponible. Les logiciels de CAO sont conçus pour l'une ou l'autre de ces approches de conception, mais la première est de loin la plus fréquente. Pour cette raison, nous avons développé le modèle du Laboratoire d'Expérimentation en Electrotechnique en considérant prioritairement les expérimentations normalisées. Dans ce cas la géométrie d'un dispositif électrique est fixe, et tous les domaines physiques possibles à étudier sont prédéfinis.

En examinant les caractéristiques de chaque composant du laboratoire, nous trouvons que le modèle du Laboratoire d'expérimentation en Electrotechnique représente un système qui se compose d'objets autonomes interagissant fortement. Que ce soit le dispositif électrique, un instrument de mesure ou une source électrique, chacun est un objet indépendant capable d'agir sur lui-même et son environnement. Ce fait nous conduit, pour la réalisation informatique du modèle (chapitre 5), à nous intéresser à l'architecture multi-agents (chapitre 4).

## Chapitre 4

# **Architecture des Systèmes Interactifs**



## Chapitre 4

### Architecture des Systèmes Interactifs

#### 4.1. Introduction

La structure d'un système interactif se définit comme l'assemblage de deux composants logiciels : l'application et l'interface interactive (ou l'interface utilisateur). L'application est un ensemble de concepts qui définissent le domaine de fonctionnement du système. L'interface interactive, quant à elle, peut se décomposer en un composant de présentation et un composant de contrôle de dialogue, ce dernier assurant la communication entre l'application et l'utilisateur.

Pour un système interactif, la modélisation des interactions est un élément fondamental, nécessaire en particulier lors du processus de développement des interfaces. Le modèle d'interaction établi facilite la compréhension de la communication homme - machine. Il existe deux approches principales utilisées pour modéliser l'interaction homme - machine : l'analyse comportementale et l'analyse structurale. La première approche propose des modèles qui décrivent directement l'interaction en termes de manipulation de l'utilisateur, de messages du système et d'enchaînement entre les séquences d'interaction. La deuxième propose des modèles qui concernent, dans le contexte architectural, les relations de l'interface utilisateur ou de l'interface homme - machine avec le reste du système interactif.

Ce chapitre est consacré aux modèles d'architecture des systèmes interactifs. Nous décrirons, d'abord, les modèles d'architecture proposés pour les systèmes interactifs. Ensuite, nous exposerons plus en détail le modèle multi-agents, en particulier le modèle AMF (Agent Multi-Facettes) qui nous intéresse plus particulièrement. Nous avons en effet constaté que le modèle AMF est en mesure de satisfaire aux exigences du système de laboratoire d'expérimentation en Electrotechnique présenté dans le chapitre précédent.

#### 4.2. Modèles d'architecture des systèmes interactifs

Un modèle d'architecture proposé pour un système interactif devrait remplir les trois objectifs suivants [Tarpin 99] :

- servir de support lors de la spécification,
- constituer l'ossature de la réalisation,
- servir de guide d'utilisation.

Les modèles d'architecture de systèmes interactifs, au regard de la structuration de leurs composants, peuvent être classifiés en trois catégories. Ce sont, soit des modèles en couches, soit des modèles multi-agents, soit des modèles mixtes. Nous allons parler des caractéristiques de chaque catégorie dans les sous chapitres suivants.

#### **4.2.1. Modèles en couches**

Les modèles en couches sont des modèles centralisés tels que le modèle *Seeheim* et le modèle *Arch*. Ils ont été développés en utilisant l'approche de l'analyse structurale.

Selon le modèle *Seeheim* un système interactif, ou une application interactive, se compose de quatre composants : le noyau fonctionnel (l'application), l'interface de l'application, le contrôle de dialogue et la présentation. Les trois derniers composants constituent l'interface utilisateur du système interactif. Le noyau fonctionnel définit les concepts correspondant au fonctionnement du système. Le contrôle de dialogue est considéré comme un médiateur entre l'interface de l'application et la présentation. La présentation, quant à elle, définit le comportement du système interactif tel qu'il est perçu et manipulé par l'utilisateur. L'interface de l'application est le représentant du noyau fonctionnel. Elle convertit les concepts du noyau fonctionnel en phrases compréhensibles par le contrôle de dialogue.

Dans le modèle *Arch*, on retrouve les composants conceptuels du modèle *Seeheim* : le noyau fonctionnel, le contrôle de dialogue et la présentation (figure IV-1). En supplément, ce modèle introduit la notion d'adaptateur de domaine et répond ainsi aux lacunes du modèle *Seeheim* qui confond la décomposition fonctionnelle et la décomposition structurale.

L'adaptateur de domaine a essentiellement pour fonction d'ajuster les différences de modèle entre les objets conceptuels du noyau fonctionnel et ceux du contrôle de dialogue. Suivant le même point de vue, le composant de présentation, peut être considéré comme l'adaptateur d'interaction qui règle les différences de modélisation entre les objets conceptuels du composant d'interaction et de ceux du contrôle de dialogue. Enfin, le composant d'interaction, en contact direct avec l'utilisateur, peut être considéré comme le constructeur des objets d'interaction.

Cette approche, certes structurante, possède cependant des restrictions importantes. D'une part, elle est très centralisée, ce qui dans le cas des grosses applications peut poser rapidement des problèmes d'évolutivité, et, d'autre part, elle ne définit pas précisément les protocoles de communication entre les composants.

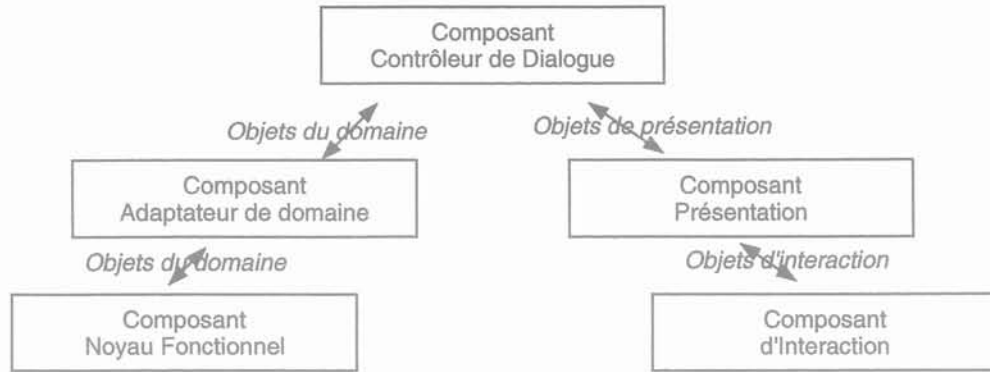


Figure IV - 1 Les composants du modèle Arch

#### 4.2.2. Modèles multi-agents

Nous commencerons l'étude du modèle multi-agents en définissant au préalable la notion d'agent. Pour cela, nous prenons la définition introduite par [Ferber 95]. Un agent est une entité informatique, réelle ou abstraite, indépendante et à durée de vie limitée, capable d'agir sur elle-même et son environnement. Ainsi, un agent se comporte comme une entité qui possède des connaissances, des initiatives et des capacités pour interagir ou communiquer avec ses pairs. Il existe trois approches différentes à propos du type d'agents qui constituent un système multi-agents.

Les personnes travaillant sur les logiciels d'agents, considèrent qu'un système est vraiment multi-agents, s'il se compose d'agents autonomes. Un agent autonome est défini, lui-même, comme un système situé dans un environnement, faisant partie de cet environnement, capable d'interagir avec son environnement, et possédant un agenda pour ses actions futures [Franklin 96].

L'agent cognitif est introduit par les chercheurs qui travaillent dans le domaine de l'intelligence artificielle. Chaque agent cognitif possède des connaissances sur un domaine particulier. Ils coopèrent pour résoudre des tâches complexes.

Dans la modélisation des interfaces homme - machine, les chercheurs considèrent que le système interactif modélisé comme un système multi-agents, est composé d'agents réactifs. Un agent réactif, ou un agent interactif, se définit comme un agent dont le comportement est la conséquence de ses observations et de ses interactions avec l'utilisateur [Tarpin 99]. Parfois, l'agent réactif peut être modélisé en prenant en compte le modèle d'agent cognitif.

La première approche permet, en particulier, de distinguer un logiciel d'agents d'un logiciel ordinaire. Cette approche décrit donc le comportement par rapport à l'environnement dans lequel il sera implanté et ses capacités à propos de son futur développement. On peut alors considérer qu'un système multi-agents est un logiciel d'agents qui contient d'autres logiciels d'agents.



L'approche agent cognitif et l'approche agent réactif, correspondent à des approches plus élémentaires que l'approche agent autonome qui intègre plus l'aspect système. Pour le logiciel, elles définissent le comportement des composants.

Les modèles multi-agents sont des modèles distribués qui utilisent l'approche de l'analyse comportementale pour le modèle d'interaction homme - machine. Ils décomposent un système interactif en plusieurs entités autonomes et coopératives. Cette décomposition convient pour **un système concerné par la manipulation de nombreux comportements particuliers et pour lequel il n'y a pas d'influence sur le fonctionnement global à prévoir** et, bien sûr, pour **un système constitué par un ensemble de composants autonomes avec beaucoup d'interactions**. De plus, grâce à l'autonomie relative des agents dans l'interprétation des événements et dans la gestion de leur état, les modèles multi-agents permettent d'avoir plusieurs dialogues en parallèle.

Ces modèles multi-agents se caractérisent par une organisation fortement modulaire, des traitements exécutés en parallèle et une communication par événements [Coutaz 88]. Si on considère la modularité et la communication, ce modèle peut être rapproché des modèles objets. En effet, un agent peut être construit comme une instance d'une classe particulière. Ceci est un des avantages des modèles multi-agents, puisqu'il existe déjà des outils facilitant leur implantation.

Par contre, l'inconvénient des modèles multi-agents est une forte dispersion de l'interface globale. La perception de l'interface globale est difficile pour le concepteur de nouvelles applications, car elle est répartie entre les différentes facettes *Présentation* de chaque agent. Cependant, nous verrons qu'à l'aide du modèle de programmation "*scene graph*" du Java3D, que nous allons décrire dans le chapitre 5, nous pouvons facilement construire une interface globale par assemblage de composants *Présentation* de chaque agent.

Un exemple représentatif de modèle multi-agents est le modèle *PAC* (Présentation Abstraction Contrôle) [Coutaz 87]. Ce modèle décompose un système interactif en une structure hiérarchique d'agents interactifs (figure IV-2). En adoptant une approche homogène, c'est-à-dire la structure de chaque agent est définie de la même manière, un agent interactif *PAC* se structure fondamentalement en 3 facettes :

- la facette *Présentation* qui prend en charge l'interface utilisateur : la transmission des informations à l'utilisateur et l'acquisition des entrées de l'utilisateur,
- la facette *Abstraction* qui définit les données et les traitements sémantiques en correspondance avec l'état ou le fonctionnement de l'agent,
- la facette *Contrôle* qui est le moteur des communications entre le composant *Présentation* et le composant *Abstraction*, et entre l'agent et les autres agents.

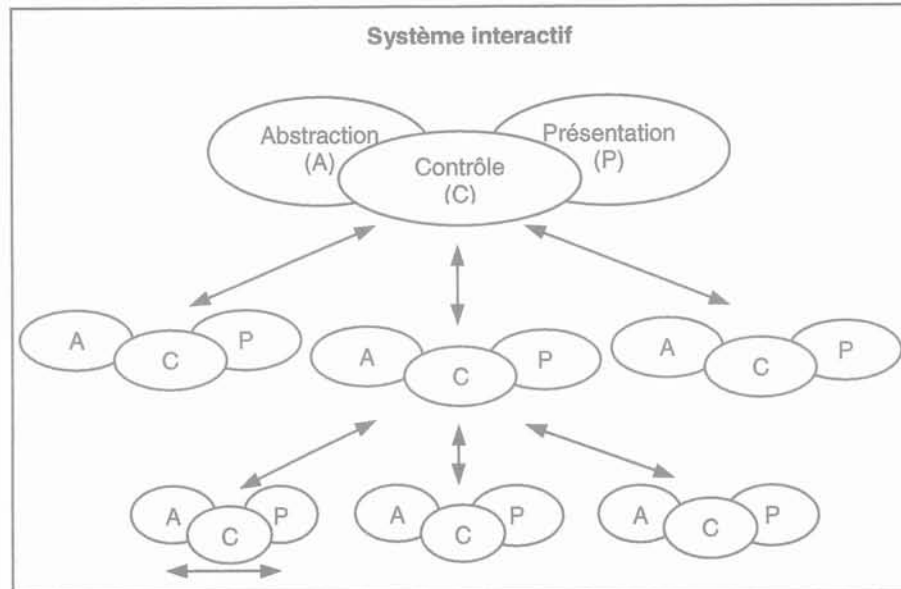


Figure IV - 2 La structure d'un système interactif PAC

Le modèle *PAC* a été développé pour répondre aux restrictions des modèles multi-agents en général qui ne fournissent aucune indication sur l'organisation des dialogues en parallèle ou des traitements de l'interaction. Dans ce modèle, la facette *Abstraction* n'a pas l'autorisation de communiquer directement avec la facette *Présentation* et inversement. La communication doit passer par la facette ou le composant *Contrôle* qui sert à exprimer la coordination et la transformation de formalisme entre des perspectives abstraite et réelle. En outre, elle gère la communication entre les agents qui s'effectue ainsi de manière hiérarchique.

#### 4.2.3. Modèles mixtes

Un exemple représentatif des modèles mixtes est le modèle *PAC-Amodeus* proposé pour des applications multi-modales. Ce modèle utilise le modèle *Arch* comme la base pour décomposer fonctionnellement un système interactif et implémente le modèle *PAC* pour structurer le contrôle de dialogue.

Conformément à l'objectif des modèles mixtes, le modèle *PAC-Amodeus* réunit les avantages des deux modèles. Le modèle *Arch* encourage l'existence de code réutilisable, correspondant aux objets d'interaction d'une boîte à outils, et sait adapter l'hétérogénéité et le changement grâce à ses deux adaptateurs. Mais, il ne fournit pas d'abstraction pour structurer son contrôle de dialogue d'une façon compatible avec les besoins des tâches de l'utilisateur. De son côté, le modèle *PAC* encourage la décomposition des tâches, mais il n'exprime pas le lien avec le code réutilisable existant.

### 4.3. Modèle AMF

L'AMF est un modèle d'architecture multi-agents multi-facettes [Ouadou 94]. Le modèle AMF, qui remplit les trois objectifs du modèle d'architecture logicielle, est inspiré par le modèle PAC. A la différence du modèle PAC, le modèle AMF propose :

- une décomposition des agents interactifs, plus fine que le modèle PAC,
- un formalisme performant de modélisation des composants *Contrôle*.

#### 4.3.1. Structure des facettes

Une facette est un composant contenant l'ensemble des données et traitements relevant d'une même thématique fonctionnelle [Tarpin 99]. Dans chaque agent interactif AMF, on peut trouver les trois facettes classiques du modèle PAC et d'autres nouvelles facettes dont le rôle est suffisamment spécifique pour justifier leur existence. Ces nouvelles facettes peuvent provenir :

- d'une duplication des facettes classiques, par exemple plusieurs facettes *Présentation*,
- de l'identification de nouveaux aspects des agents, par exemple : une ou plusieurs facette(s) *Erreur* pour la gestion des erreurs (détection, signalement et correction), une ou plusieurs facette(s) *Aide* pour fournir une aide sur les caractéristiques des agents, etc.

Le nombre des facettes dans chaque agent n'est pas limité, et aucune facette n'est obligatoire.

Comme dans le modèle PAC, les agents interactifs sont reliés dans une structure hiérarchique selon des règles de composition qui traduisent qu'un agent parent contient un certain nombre d'agents enfants ou d'agents fils. Pour le modèle AMF, la relation hiérarchique des agents est représentée par une représentation graphique emboîtée ou une représentation rationnelle (figure IV- 3).

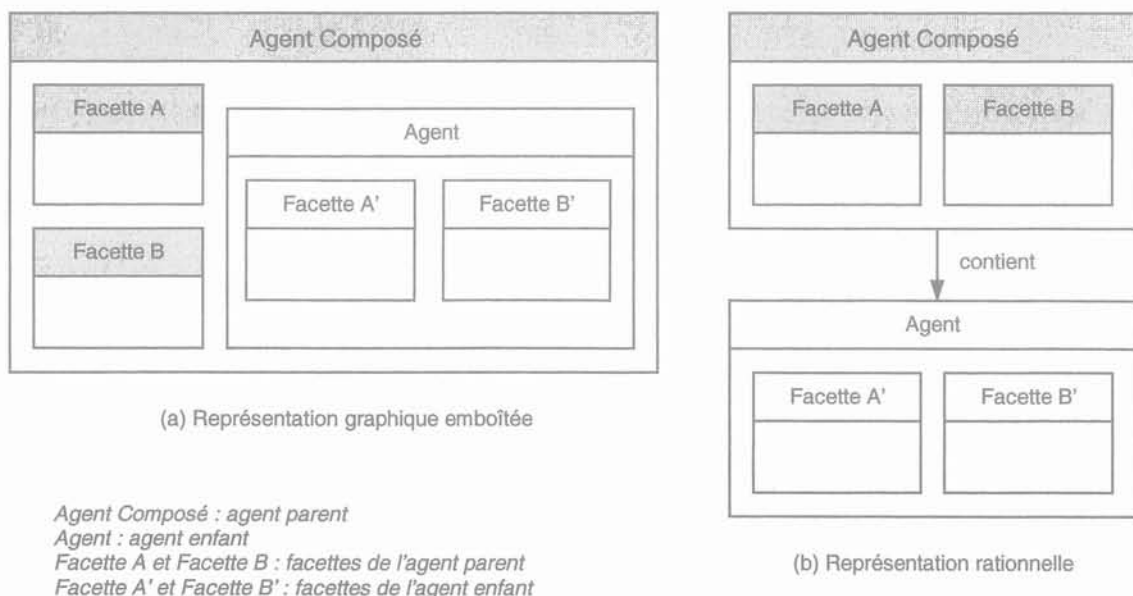


Figure IV - 3 Représentation de la relation hiérarchique des agents AMF

### 4.3.2. Composants de la communication

Comme nous l'expliquions précédemment, la facette *Contrôle* gère les communications entre les facettes propres d'un agent et entre les agents. Le formalisme de cette facette est développé en considérant qu'elle doit être capable de traduire des relations de type : "la facette A veut utiliser la fonction  $f$  de la facette B".

Concernant le type des relations qui devrait être rempli, les communications entre les facettes d'un même agent ou de différents agents s'effectuent par l'intermédiaire de deux types d'éléments principaux :

- les ports de communication, situés au niveau des facettes, qui sont les interfaces des facettes et définissent les fonctions proposées et utilisées,
- les administrateurs de contrôle, situés au niveau des agents, qui sont les composants essentiels de la facette *Contrôle* et établissent des liens entre les ports de communication.

#### 4.3.2.1. Les ports de communication

Il existe trois types de ports de communication : **port d'entrée** ou port cible, **port de sortie** ou port source et **port d'entrée/ sortie**. Un port d'entrée offre une fonction ou un service. Un port de sortie est un port qui a besoin d'un service. Le port d'entrée/sortie se comporte, tantôt comme un port d'entrée, tantôt comme un port de sortie.

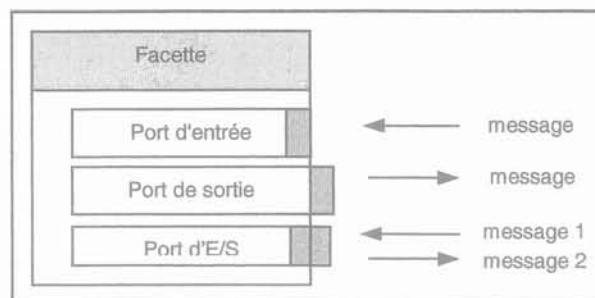


Figure IV - 4 Représentation des ports de communication d'une facette AMF

Une fonction particulière, appelée *démon*, peut être associée à chaque port de communication. Cette fonction se déclenche automatiquement lorsqu'un port est activé. Les mécanismes de déclenchement des démons sont décrits dans le tableau IV-1.

Pour faciliter les communications entre les agents, un port de sortie peut accéder à un service fourni par d'autres agents, et un port d'entrée peut être activé par d'autres agents. C'est aussi vrai pour un port d'entrée/ sortie. Dans ce cas, les ports sont dits *exportés*. Les ports de communication *exportés* sont situés au niveau des agents. Ils sont indispensables, car ils constituent l'interface fonctionnelle visible de l'agent. Une exportation est représentée sur la représentation graphique par un petit cercle situé à la frontière des agents (figure IV-5).

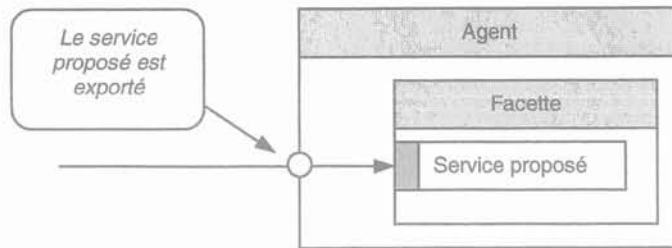
Figure IV - 5 Représentation d'un port de communication *exporté*

Tableau IV - 1 Les situations d'activation des démons

Port actif	Situations
Port d'entrée	Le "démon" associé remplit effectivement le service offert. Il traite les informations contenues dans le message accepté par le port
Port de sortie	<p>Souvent, il n'y a pas de démon particulier qui est associé à ce port. L'objectif de ce port est de servir de point de départ pour l'invocation d'un service distant ou d'envoyer un message.</p> <hr/> <p><b>note</b> : dans certains cas, ce port peut contenir un démon particulier pour faire des pré-traitements avant d'envoyer le message</p>
Port d'entrée/sortie	<p>Première activation : ce port se comporte tout d'abord comme un port d'entrée.</p> <p>Deuxième activation : ce port se comporte comme un port de sortie</p> <hr/> <p><b>note</b> : le démon associé à ce port est activé lors de la première activation.</p>

#### 4.3.2.2. Les administrateurs de contrôle (les facettes Contrôle)

Les administrateurs de contrôle sont les éléments qui constituent la facette *Contrôle*. A la différence des autres facettes, elle n'a pas véritablement d'existence propre. C'est-à-dire que pour sa réalisation, la

facette *Contrôle* n'est pas définie comme une classe particulière, mais elle est représentée par des classes des administrateurs de contrôle qui seront implémentées par des classes des agents. La figure IV-6 montre la représentation graphique de la facette *Contrôle* constituée par l'administrateur de contrôle dans une relation indiquant que la facette A veut utiliser le service *f* de la facette B.

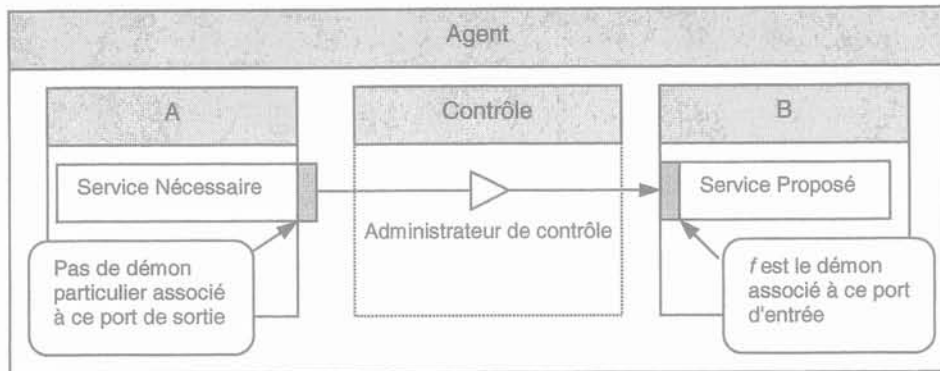


Figure IV - 6 Modélisation d'un administrateur de contrôle

Les administrateurs de contrôle qui gèrent les liens entre les ports sources et les ports cibles des facettes sont chargés de trois rôles :

- un rôle de connexion, ils gèrent les relations logiques entre les ports sources et les ports cibles,
- un rôle de traduction, ils transforment les valeurs des ports sources en valeurs compréhensibles par les ports cibles,
- un rôle comportemental, ils savent traduire les règles d'activation des ports cibles.

En correspondance avec le rôle de traduction, le modèle *AMF* introduit les types d'administrateurs présentés sur le tableau IV-2. En plus, pour le rôle comportemental, ce modèle propose les types d'administrateurs décrits par le tableau IV-3. L'administrateur de séquence, dans le cas de l'activation sélective de certains ports cibles, est aussi appelé administrateur filtrant, car il n'active pas tous les ports cibles, mais uniquement ceux à qui est destiné le message. Les administrateurs comportementaux permettant le mécanisme de retour, sont classifiés comme les administrateurs avec retour, car ils activent un port cible et retournent une réponse au(x) port(s) source(s) d'où les messages proviennent.

En définitive, un administrateur de contrôle contient un administrateur comportemental et un administrateur de traduction.

Tableau IV - 2 Les administrateurs de traduction







Type	Mécanisme de traduction	Représentation schématique
Administrateur de <i>transfert</i>	Transporte directement les données du port source vers le port cible, sans modification	
Administrateur de <i>conversion</i>	Transforme les données issues du port source en données compréhensibles par le port cible	
Administrateur de <i>correspondance</i>	Effectue une correspondance entre l'ensemble des valeurs des ports sources et celui des valeurs du port cible	
Administrateur d' <i>assemblage</i>	Rassemble les données issues des ports sources pour les transporter sous une forme structurée vers le port cible	
Administrateur de <i>calcul</i>	Applique des opérations de calcul sur les données issues des ports sources pour construire les données à envoyer au port cible	
Administrateur de <i>traitement</i>	Applique un algorithme sur les données issues des ports sources pour obtenir les données à envoyer au port cible	

Tableau IV - 3 Les administrateurs comportementaux

Type	Règle d'activation des ports cibles	Représentation schématique
Administrateur <i>simple</i>	Si port source A est activé alors activer port cible B	
Administrateur <i>itératif</i>	Représente une répétition de n activations du port source avant d'avertir le port cible	
Administrateur de <i>séquence</i>	Représente une relation d'attente : <ul style="list-style-type: none"> <li>- entre plusieurs ports sources (en certain ordre) et un port cible : <i>si port source A1 est activé puis port source A2 puis .... port source An, alors activer port cible B</i></li> <li>- entre un port source et plusieurs ports cibles(en certain ordre) : <i>si port source A est activé, alors activer port cible B1 puis port cible B2 puis .... Port cible Bn</i></li> </ul>	
Administrateur <i>conjonctif</i>	Représente une relation de conjonctif sans contrainte d'ordre: <i>Si port source A1 et port source A2 et ... port source An, alors port cible B</i>	
Administrateur <i>disjonctif</i>	Représente une relation de disjonctif : <i>Si port source A1 ou port source A2 ou ... port source An, alors port cible B</i>	

#### 4.3.2.3. Le mécanisme de communication

Le mécanisme de communication dans le modèle *AMF* est assimilé au mécanisme d'activation des ports de communication, des agents et des administrateurs de contrôle. Il s'effectue comme suit :

1. Lorsqu'un port de communication étant un port source est activé :
  - Le *démon* associé se déclenche
  - L'activation est transmise à l'agent
  - L'agent réveille tous les administrateurs de contrôle sur lesquels le port est connecté



- Si le port est exporté, l'activation est aussi transmise récursivement à d'autres agents cibles
2. Chaque administrateur de contrôle ainsi réveillé examine ses conditions d'activation :
    - Si les conditions sont remplies, il traduit les données ou les messages se trouvant sur chacun des ports sources actifs pour construire un nouveau message
    - Le nouveau message est envoyé à tous les ports cibles en fonction du rôle comportemental de l'administrateur.
    - Si les ports cibles se situent sur d'autres agents, l'activation est propagée à ces agents
    - L'activation des ports cibles est ensuite suivie par le déclenchement des *démons* associés
  3. En particulier, si l'administrateur de contrôle fournit le mécanisme de retour :
    - Les ports cibles renvoient une réponse ou un message d'accusé de réception à l'administrateur
    - L'administrateur retourne un tel message au port source qui a déclenché l'activation.

### 4.3.3. Application

La mise en œuvre du modèle *AMF* se fait en deux phases : une phase de construction de l'architecture et une phase d'utilisation de l'architecture construite. La phase de construction consiste à construire les agents composés, les agents composants, les facettes, les administrateurs de contrôle, les ports sources et les ports cibles du système interactif modélisé, et aussi à définir les liens entre ces ports sources et ces ports cibles. La deuxième phase consiste à définir le mécanisme de communication entre les composants construits. Elle correspond à l'activation d'un port de sortie, au déclenchement de son *démon*, à l'émission d'un message vers les ports cibles et aux déclenchements des *démons* associés aux ports cibles.

La figure IV-7 présente la succession des primitives durant les deux phases de l'application du modèle *AMF*. Les opérations pendant la première phase correspondent aussi aux étapes que l'on fait lors de la construction de la racine des composants en programmation orientée objet. Ainsi, comme l'*AMF* a été développé pour qu'il puisse remplir les objectifs du modèle d'architecture logicielle, un schéma de modélisation *AMF* (la représentation graphique *AMF*), peut se transposer facilement lors de la programmation des composants du système interactif.



Figure IV - 7 La succession des primitives de l'application du modèle AMF

#### 4.4. Conclusion

Comme nous l'avons décrit dans le chapitre précédent, le Laboratoire d'Expérimentation en Electrotechnique est un système réel constitué par un ensemble des composants autonomes qui interagissent. Dans le chapitre suivant, nous allons constater que l'approche multi-agents est particulièrement bien adaptée au développement d'un système interactif tel que le modèle informatique du Laboratoire d'Expérimentation en Electrotechnique.

L'approche multi-agents a déjà permis le développement de systèmes comparables en complexité au système du Laboratoire d'Expérimentation en Electrotechnique. Par exemple, cette approche a été adoptée par [Servat 98] pour construire un système de simulation des phénomènes complexes dans lequel se réalise le processus de synthèse mettant en œuvre de très nombreux événements interactifs. Une autre application de cette approche a été introduite par Chevaillier. Il l'a implantée en Réalité

Virtuelle pour créer un logiciel qui ne peut pas être décrit comme un seul système monolithique, mais comme un ensemble de composants autonomes interagissant fortement [Chevaillier 99]. Une simulation informatique de mondes, constitués par un ensemble de processus parallèles interagissant à travers la production et l'interprétation de messages, s'appuie également sur les systèmes multi-agents [Treuil 97].

Parmi les modèles multi-agents présentés, nous avons choisi le modèle *AMF*, car il nous permet d'avoir les avantages du modèle *PAC*, c'est-à-dire l'indépendance entre l'abstraction et la présentation. Cela autorise la modification locale à tous les niveaux, sans mettre en cause le fonctionnement global. De plus, ce modèle permet de décomposer les agents interactifs de façon plus fine que le modèle *PAC* et fournit un formalisme performant pour modéliser la facette *Contrôle*. Comme le modèle *AMF* a été développé en remplissant les objectifs du modèle d'architecture logicielle, se reporter à 4.2, il permet de transmettre facilement son schéma de modélisation au niveau de la programmation des objets du système interactif (lors du codage Java). En outre, ce schéma représente le modèle d'interaction homme - machine, qui englobe les interactions de l'utilisateur et les interactions internes, aussi bien que la structuration hiérarchique des composants du système interactif.

Le développement de notre système interactif, appelé *Laboratoire Virtuel d'Expérimentation en Electrotechnique*, sera présenté dans le chapitre 5. Pour remplir le besoin particulier de ce système interactif, nous avons adopté le modèle *AMF* en créant les comportements adaptés de l'administrateur de contrôle.

## Chapitre 5

# **Laboratoire Virtuel d'Expérimentation en Electrotechnique (LVEE)**



## Chapitre 5

### Laboratoire Virtuel d'Expérimentation en Electrotechnique (LVEE)

#### 5.1. Introduction

Le Laboratoire Virtuel d'Expérimentation en Electrotechnique (LVEE) est un système interactif représentant le Laboratoire d'Expérimentation en Electrotechnique que nous avons modélisé dans le chapitre 3. En ce qui concerne l'implémentation de la technologie de Réalité Virtuelle (RV), le LVEE est développé comme un système de RV sans immersion. Les mondes virtuels du LVEE sont visualisés sur un écran ordinaire avec utilisation de la technologie d'interface classique : un clavier et une souris 2D. Cet équipement, bien évidemment, ne permet pas de donner à l'utilisateur l'impression d'être immergé dans ces mondes virtuels (§ 2.2.2).

Le LVEE est conçu comme l'interface d'exploitation des logiciels de CAO en Génie Electrique. Au sein d'un système de CAO, le LVEE fonctionne comme un système de gestion des données. Ce système interactif est réalisé en utilisant les API du langage de programmation Java pour définir les comportements des composants le constituant et les API Java3D pour créer des présentations visuelles de ces composants.

Dans ce chapitre, nous parlerons, tout d'abord, des étapes de développement du LVEE en nous référant à la méthode de conception d'un système interactif dite du "*Cycle en V*". Ensuite, nous présenterons les spécifications du LVEE. Elles seront suivies par quelques explications à propos de son architecture. En effet, le LVEE est constitué d'agents interactifs qui sont définis comme des classes en Java et Java3D. Les classes de ces agents interactifs seront expliquées puis viendront quelques éclaircissements sur la présentation visuelle, les interactions avec les données et le pilotage des logiciels de simulation.

## 5.2. Développement

En s'appuyant sur la méthode de conception d'un système interactif du "*Cycle en V*" [Coutaz 98], présentée dans l'annexe L, le LVEE est développé en construisant tout d'abord le modèle du Laboratoire d'Expérimentation en Electrotechnique. Cela englobe les modèles des tâches et des interactions. Ensuite, nous examinons les caractéristiques de ce modèle afin de déterminer l'architecture logicielle qui est capable de l'accueillir.

L'étape suivante est la spécification de la technologie d'interface. Ces spécifications nous permettent, entre autres, de décider des langages de programmation utilisés pour la réalisation informatique du système. Pour prévoir l'implémentation de la technologie de Réalité Virtuelle (RV), nous avons décidé d'utiliser l'API Java3D (§ 2.4.2) pour, en particulier, définir la présentation visuelle du LVEE. Parmi les points forts de cet API, son modèle de programmation "*scene graph*" est particulièrement bien adapté à la construction de la présentation globale du LVEE qui est un assemblage des composants *Présentation* de chaque agent interactif, ainsi qu'à la visualisation des dispositifs électriques qui sont décrits par décomposition structurelle. Cependant, Java3D ne supporte pas directement tous les aspects possibles pour une application en 3D. Pour le compléter nous utilisons aussi des parties écrites en Java.

Après la construction de l'architecture et la réalisation informatique, qui dépendent des spécifications du LVEE, un test d'acceptation est fait pour vérifier l'utilisabilité de ce système.

## 5.3. Spécification

La conception des dispositifs électriques étant de plus en plus précise, les logiciels de CAO en Génie Electrique deviennent de plus en plus complexes, car ils doivent tenir compte de tous les domaines physiques susceptibles d'être étudiés sur les dispositifs (§1.3). Par rapport aux domaines d'application, ces logiciels devraient faciliter aussi la compréhension des concepts scientifiques intervenant sur le fonctionnement des dispositifs (§ 2.5.2) et la construction d'une simulation interactive permettant d'étudier un système électrique de manière réaliste (§ 2.5.3). Voici donc ce que nous attendons du LVEE :

- Le LVEE sera une interface d'aide à l'exploitation des logiciels de CAO en Génie Electrique. A travers des interactions entre l'utilisateur et les données, similaires aux activités d'expérimentation de dispositif électrique, le LVEE devra fournir les paramètres d'entrée nécessaire aux logiciels de simulation, contrôler le processus de simulation et interpréter les résultats de simulation.
- Le LVEE devra avoir une présentation visuelle proche de celle du Laboratoire d'Expérimentation en Electrotechnique pour que le savoir-faire, acquis pendant la simulation avec des logiciels de CAO, soit directement transférable dans le monde réel. Cela peut faciliter, en particulier, l'indispensable phase de validation expérimentale de la modélisation numérique [Oren 78].
- Le LVEE devra proposer une visualisation naturelle afin de faciliter la compréhension des phénomènes physiques qui influent sur le fonctionnement des dispositifs électriques. Le modèle du

dispositif aura, alors, le même comportement que son homologue réel. En plus, il sera souhaitable, à terme, d'observer les événements se produisant à l'intérieur des dispositifs et de ressentir ceux qui n'ont pas de présentation visuelle.

- Le LVEE devra pouvoir fournir des informations nécessaires pour analyser les problèmes observés sur les dispositifs électriques réels.

## 5.4. Architecture

Basé sur le modèle AMF (§ 4.3), que nous avons décrit précédemment, le LVEE se compose d'agents interactifs qui sont liés dans une structure hiérarchique.

### 5.4.1. Structure hiérarchique des agents interactifs

La structure hiérarchique des agents interactifs du LVEE (figure V-1) est inspirée de la structure des composants du Laboratoire d'Expérimentation en Electrotechnique. Le LVEE est, en effet, représenté par l'agent *"VELaboratory"* qui est l'agent parent des autres agents interactifs. Au premier niveau de cette hiérarchie, on trouve :

- L'agent *"Cupboard"*;  
Cet agent définit le monde virtuel de l'armoire qui contient des informations à propos des types des dispositifs électriques, des objectifs d'expérimentation et des conditions d'expérimentation de chaque objectif d'expérimentation.
- L'agent *"Experiment"* ;  
Cet agent représente le monde virtuel de la salle dans laquelle des expérimentations sont effectuées.

Ensuite, en se référant aux entités d'une expérimentation de dispositif électrique, l'agent *"Experiment"* est découpé en agents fils :

- L'agent *"Device"* ;  
Cet agent réunit les différentes catégories de dispositifs électriques. Il sait à quelle catégorie le dispositif choisi appartient.
- L'agent *"Montage"*;  
Il regroupe des agents fils qui sont chargés de calculer des comportements physiques des dispositifs électriques.

L'agent *"Device"* lui-même se décompose donc en agent *"Contactor"*, agent *"Transformer"*, agent *"MachineDC"* et agent *"MachineAC"*. Chacun gère le type de contacteur, de transformateur, de machine à courant continu et de machine à courant alternatif qui doit être construit selon le choix de l'utilisateur. Ils se composent eux-mêmes des agents interactifs représentant des sous-types de dispositifs électriques. Par exemple, l'agent *"Contactor"* est constitué des agents des contacteurs possédant un électroaimant en forme de U, en forme de E et en noyau plongeant. L'agent interactif d'un dispositif électrique devrait définir la présentation géométrique et les comportements physiques que possède le vrai dispositif.



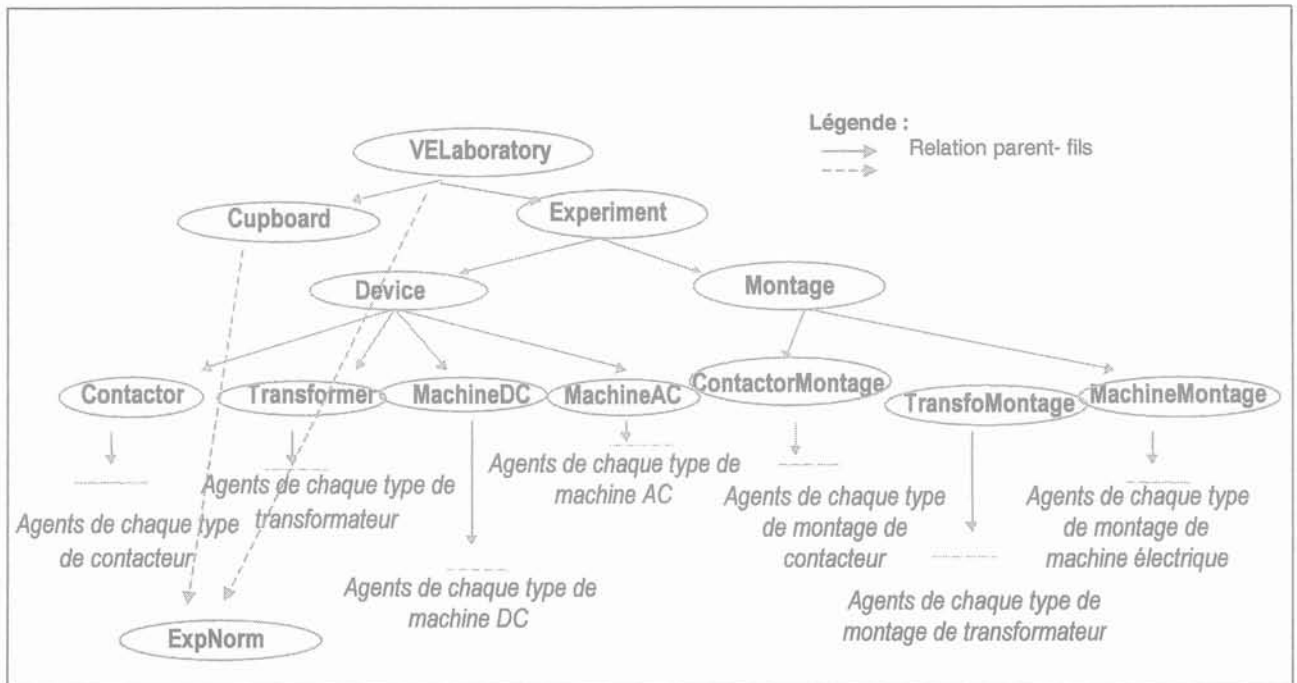


Figure V - 1 Structure hiérarchique des agents interactifs du LVEE

De la même manière, l'agent "*Montage*", est découpé en des agents interactifs qui, chacun, organise des montages d'expérimentation pour chaque catégorie des dispositifs électriques. Il s'agit de l'agent "*ContactorMontage*", de l'agent "*TransfoMontage*" et de l'agent "*MachineMontage*". Ensuite, l'agent "*ContactorMontage*" se décompose en des agents qui définissent des montages d'expérimentation pour les contacteurs, il en est de même avec l'agent "*TransfoMontage*" pour les transformateurs et avec l'agent "*MachineMontage*" pour les machines à courant continu et à courant alternatif. En principe, ces agents interactifs s'occupent des calculs des comportements physiques des dispositifs électriques.

Les normes d'expérimentation des objectifs d'expérimentation sont organisées par l'agent "*ExpNorm*". Cet agent gère la norme d'expérimentation qui doit être en vigueur pour l'objectif d'expérimentation choisi. Cet agent peut être rattaché à des agents qui, pour achever leurs tâches, ont besoin d'informations sur une norme d'expérimentation.

La structuration des agents interactifs du LVEE est prévue pour que l'on puisse facilement ajouter, supprimer ou modifier les agents, à un niveau quelconque, sans remettre en cause le fonctionnement global.

#### 5.4.2. Fonctionnalités des agents interactifs

Comme nous l'avons décrit ci-dessus, chaque agent interactif du LVEE a des fonctionnalités particulières. Ces fonctionnalités sont apportées par les composants le constituant : la facette *Présentation*, la facette *Abstraction* et la facette *Contrôle* (§ 4.2.2). Le composant *Présentation* est chargé

de définir la présentation visuelle ou physique de l'agent. Ensuite, le composant *Abstraction* a pour rôle de déterminer la façon dont cet agent réagit à lui-même et à son environnement. C'est dans ce composant que les comportements et l'état de l'agent sont définis. Enfin, le composant *Contrôle* gère la communication entre les composants précédents et avec les autres agents.

Par exemple, pour l'agent interactif du contacteur avec l'électroaimant en forme E, son composant *Présentation* visualise la présentation géométrique du contacteur et anime les comportements physiques. Son composant *Abstraction* décide des réactions qui doivent être visualisées chaque fois qu'un traitement est effectué sur le contacteur. Cette décision est transmise par son composant *Contrôle* au composant *Présentation* et à d'autres agents, au cas où ces réactions affecteraient l'environnement.

L'annexe F présente, avec plus de détails, les fonctionnalités des agents interactifs du Laboratoire Virtuel d'Expérimentation en Electrotechnique.

#### **5.4.3. Communication entre les agents interactifs**

Le mécanisme de communication entre les agents interactifs du LVEE utilise la structure hiérarchique de ses agents :

- *Entre un agent parent et un agent fils juste au-dessous ;*  
Le message développé par l'agent parent est directement envoyé à l'agent fils qui est la destination unique du message (figure V-2.a).
- *Entre un agent fils et l'agent parent juste au-dessus ;*  
Inversement, le message construit par l'agent fils est directement envoyé à l'agent parent (figure V-2.b). Dans ce cas, la destination unique du message est donc l'agent parent.
- *Entre un agent parent et un agent au-dessous d'un de ses agents fils ;*  
Le message est envoyé d'abord à son agent fils. Ensuite, à partir de l'agent fils, ce message est transmis jusqu'à l'agent étant la dernière destination du message (figure V-2.c). L'agent émetteur du message doit définir tous les agents qui transmettent ce message comme des destinations intermédiaires.
- *Entre un agent et un agent au-dessous d'un autre agent ;*  
Dans ce cas, le message est envoyé à l'agent parent juste au-dessus de l'émetteur. Cet agent vérifie si le parent de l'agent destination est son fils ou se trouve dans une de ses branches. Si ce n'est pas le cas, l'envoi au niveau du dessus est répété. Par cette procédure, le message est transmis jusqu'au premier agent qui dans la hiérarchie est commun à l'émetteur et au destinataire. Alors, le message peut être envoyé dans la direction de l'agent destination (figure V-2.d). De la même manière que dans la communication précédente, tous les agents par lesquels passe le message doivent être définis comme des destinations intermédiaires.

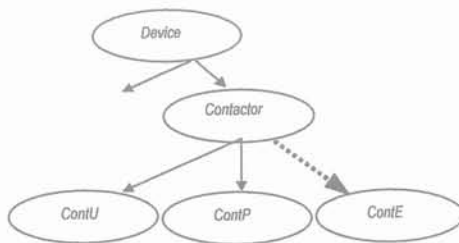
- *Entre un agent et un autre agent au-dessus de son agent parent*

Le mécanisme de communication est un cas particulier de la communication précédente. Dans ce cas, on parle d'un agent fils au milieu de l'agent parent (figure V-2.e).

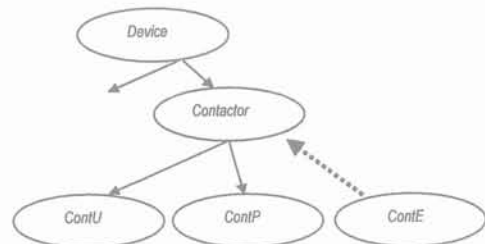
- *Entre agents au même niveau*

C'est aussi un cas particulier du mécanisme général. Le message établi est envoyé à l'agent parent qui est aussi parent de l'agent destination. Ensuite, à partir de cet agent parent le message est directement transmis à l'agent destination (figure V-2.f).

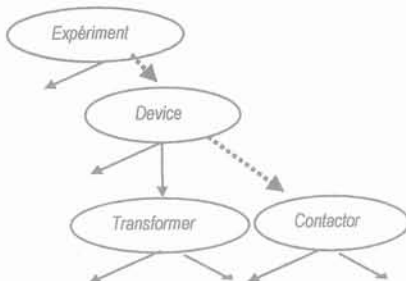
Par ailleurs, le port *importé* qui peut accéder à un service fourni par d'autres agents est redirigé vers l'agent contenant le port attaché à ce service. De même le port *exporté* est redirigé vers l'agent qui l'activera.



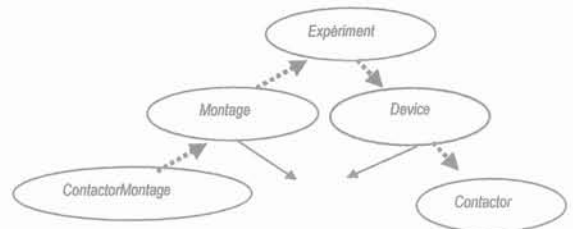
a. Entre agent parent et agent fils juste au-dessous



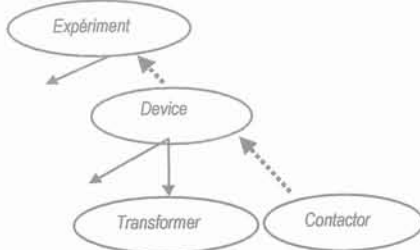
b. Entre agent fils et agent parent juste au-dessus



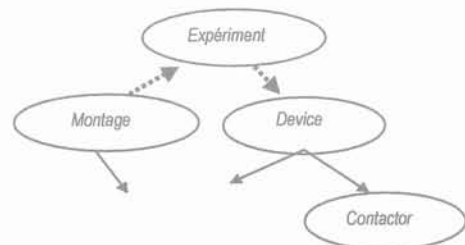
c. Avec un agent au-dessous son agent fils



d. Avec un agent au-dessous d'un autre agent



e. Avec un agent au-dessus de son agent



f. Avec un agent au même niveau

→ relation parent - fils  
 -.-> flot de messages

Figure V - 2 Mécanisme de la communication entre les agents interactifs

#### 5.4.4. Base de données

Le LVEE comprend des bases de données qui stockent des matériaux et des informations nécessitées par l'agent "ExpNorm" pour créer une norme d'expérimentation (§ annexe G).

##### 5.4.4.1. Normes d'expérimentation

Une norme d'expérimentation est définie comme un objet qui contient d'autres objets mémorisant les hypothèses, les contraintes, la charge, le régime de fonctionnement, protocole d'expérimentation et la plaque signalétique. Chacun de ces objets, sauf la plaque signalétique, utilise des informations de la base de données des normes d'expérimentation, qui sont maintenues dans le fichier *ExpStandard.dat*. La plaque signalétique, de son côté, rassemble les caractéristiques de fonctionnement introduites par l'utilisateur lors de la définition des paramètres du dispositif électrique examiné.

La base de données des normes d'expérimentation est développée à l'aide de la classe *ExperimentStandardCreationMenu*. Cette classe affiche une fenêtre pour introduire les données d'une norme d'expérimentation. Elle utilise la classe *CollectionExperimentStandard* qui crée le fichier *ExpStandard.dat*, au cas où-t-il ne serait pas encore disponible, et ajoute les données ainsi introduites.

La création d'un objet de la norme d'expérimentation est faite par la classe *ListExperimentStandard*, à l'aide de la classe *CollectionExperimentStandard*. Dans ce cas, la classe *CollectionExperimentStandard* est chargée de lire les données qui correspondent à la norme d'expérimentation concernée, dans le fichier *ExpStandard.dat*.

##### 5.4.4.2. Matériaux

De la même manière que la norme d'expérimentation, un matériau est défini par un objet de la classe *Material*. Cette classe fournit les paramètres qui correspondent aux caractéristiques magnétiques, mécaniques et thermique des matériaux, en général.

Les matériaux et leurs paramètres sont maintenus dans le fichier *Material.dat*. La classe *ListMaterialMenu* implantant la classe *MaterialDataCollection* permet de lire les matériaux existants dans la base de données et, aussi, de créer un nouveau matériau.

#### 5.4.5. Le LVEE par rapport aux logiciels de CAO

Pour décrire la position du LVEE par rapport aux logiciels de CAO en Génie Electrique, nous nous référons à l'architecture des logiciels de simulation proposée par [Leathrum 00] . Dans ce cas, un logiciel de simulation est décomposé en deux composants principaux : la gestion de données et le code de simulation qui exécute la simulation numérique (figure V-3). La gestion de données, elle-même, comprend les composants suivants :

- *Interface utilisateur* (interface d'entrée et interface de sortie);  
Ce composant permet à l'utilisateur de créer et d'interagir avec les données nécessaires pour conduire la simulation, contrôler le processus de simulation et visualiser les résultats de simulation.
- *Gestion de données interne*;  
Ce composant gère les données générées au cours de la réalisation des scénarios de simulation.
- *Analyse des paramètres d'entrée*;  
Ce composant apporte des mécanismes d'analyse concernant la distribution et de la visualisation des paramètres d'entrée.
- *Base de données*;  
Ce composant permettant la gestion générale des données est fondamental dans un système de simulation.
- *Analyse des paramètres de sortie*;  
Ce composant transforme les résultats de simulation afin qu'ils puissent avoir une forme plus compréhensible pour l'utilisateur.

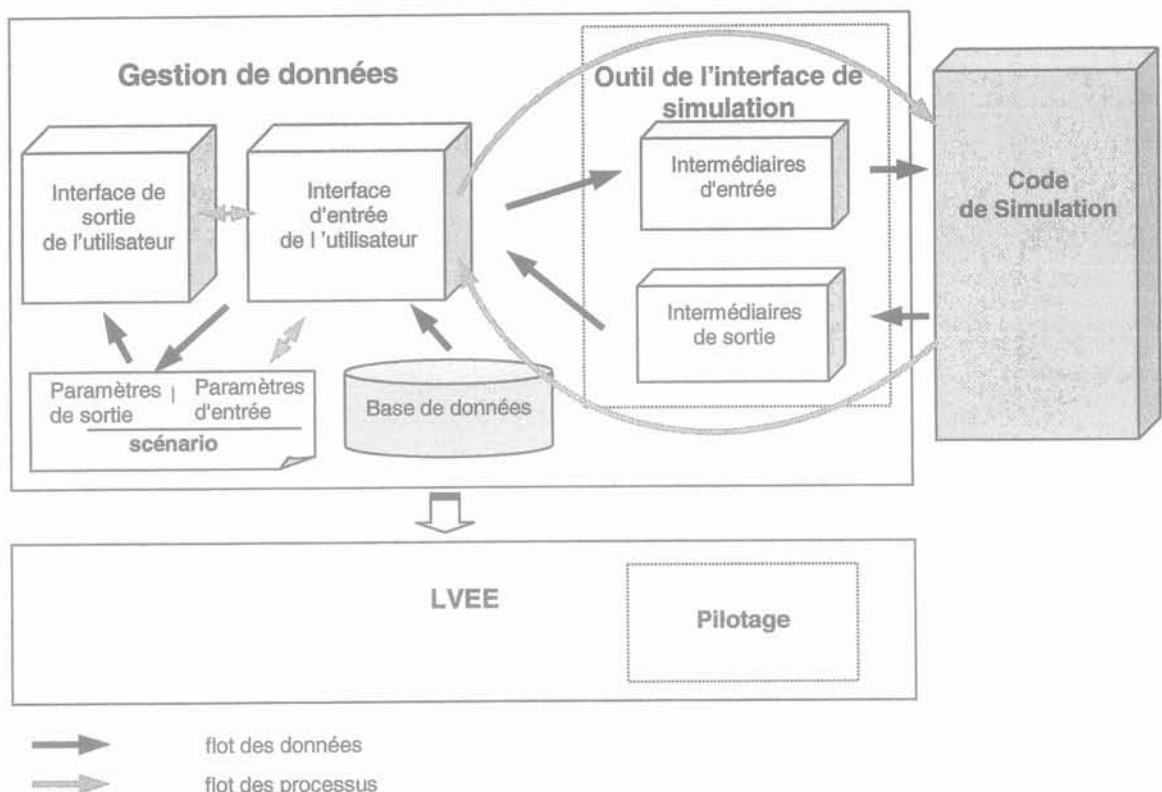


Figure V - 3 Le LVEE par rapport à l'architecture des logiciels de CAO

Développé comme une interface d'aide à l'exploitation des logiciels de CAO en Génie Electrique, le LVEE sert à la gestion des données de ces logiciels.

Le LVEE permet à l'utilisateur, à travers ses interactions avec les objets dans les mondes virtuels, de définir les données nécessaires aux logiciels de CAO et de gérer le processus de simulation et, même, de déterminer lequel, parmi les logiciels de CAO disponibles, va effectuer la simulation numérique. Les données ainsi définies sont principalement organisées par le composant *Abstraction* de chaque agent interactif. A part ces composants *Abstraction*, le LVEE contient un module chargé de la centralisation des données générées au cours de l'expérimentation, afin de faciliter le mécanisme de communication entre les agents interactifs. Le composant *Présentation* de chaque agent interactif s'occupe de la visualisation des paramètres d'entrée et de sortie. Ces paramètres sont, auparavant, analysés par le composant *Abstraction*.

Si l'on considère l'ensemble des bases de données présentes, les bases des données du LVEE sont séparées de celles des logiciels de CAO. L'accès aux bases de données des logiciels de CAO est fait à l'aide des propres outils de communication de ces logiciels (§ 5.7.1). C'est l'interface de simulation qui sert d'intermédiaire pour transmettre les données du LVEE aux logiciels de CAO et inversement, comme il est présenté sur la figure V-3.

#### 5.4.6. Scénario de simulation

Les scénarios de simulation du LVEE sont maintenus par un composant de la gestion des données interne constitué des classes statiques : *Parameter*, *Command* et *Result*. La classe *Parameter* est chargée de maintenir les informations concernant les paramètres d'entrée tels que la catégorie du dispositif examiné, le nom du dispositif et les paramètres du dispositif, etc. La classe *Command* sert à récupérer les commandes comme par exemple la création de dispositif ou du montage d'expérimentation, la disparition ou l'affichage des composants de dispositif, etc. Enfin, la classe *Result* contient les résultats de simulation numérique obtenus par les logiciels de simulation.

Le scénario de simulation créé dans le LVEE est sauvegardé dans trois fichiers : le fichier <nom\_expérimentation> qui contient les paramètres d'entrée et ses valeurs, le fichier <nom\_expérimentation>.com qui contient les commandes et le fichier <nom\_expérimentation>.res qui contient les résultats de simulation numérique.

Lorsque les valeurs des paramètres d'entrée sont introduites, les composants *Présentation* dans lesquels sont définis ces paramètres d'entrée envoient ces informations aux composants *Abstraction*. Ces derniers composants les utilisent pour mettre à jour l'état des agents auxquels ils appartiennent, avant de les stocker dans la classe *Parameter*. C'est de la même manière que les commandes sont entrées dans la classe *Commande* chaque fois qu'elles sont créées. Par contre, les résultats de simulation numérique sont directement mis dans la classe *Result* par le composant *Abstraction* de l'agent "VELaboratory". La centralisation de ces informations, en particulier celle concernant les paramètres d'entrée, a pour objectif de diminuer les temps de réponse. En effet, les communications, entre des agents éloignés dans la structure hiérarchique, sont relativement de longues durées. La diminution du

temps de communication est importante pour préserver la possibilité de réponse en temps réel. Cependant cette centralisation introduit de la complexité au niveau de la programmation, car chaque agent interactif doit connaître le nom de tous les paramètres d'entrées, de toutes les commandes et de tous les résultats de simulation numérique.

Les noms des informations créées au cours des interactions dans les mondes virtuels du LVE sont décrits dans l'annexe M. Pour l'instant, au niveau de dispositif électrique, il n'existe que les informations correspondant aux contacteurs.

## 5.5. Visualisation

La présentation visuelle du LVEE comporte le monde virtuel de l'armoire et le monde virtuel de la salle d'expérience, comme on peut le voir sur la figure V-4 ci-dessous.

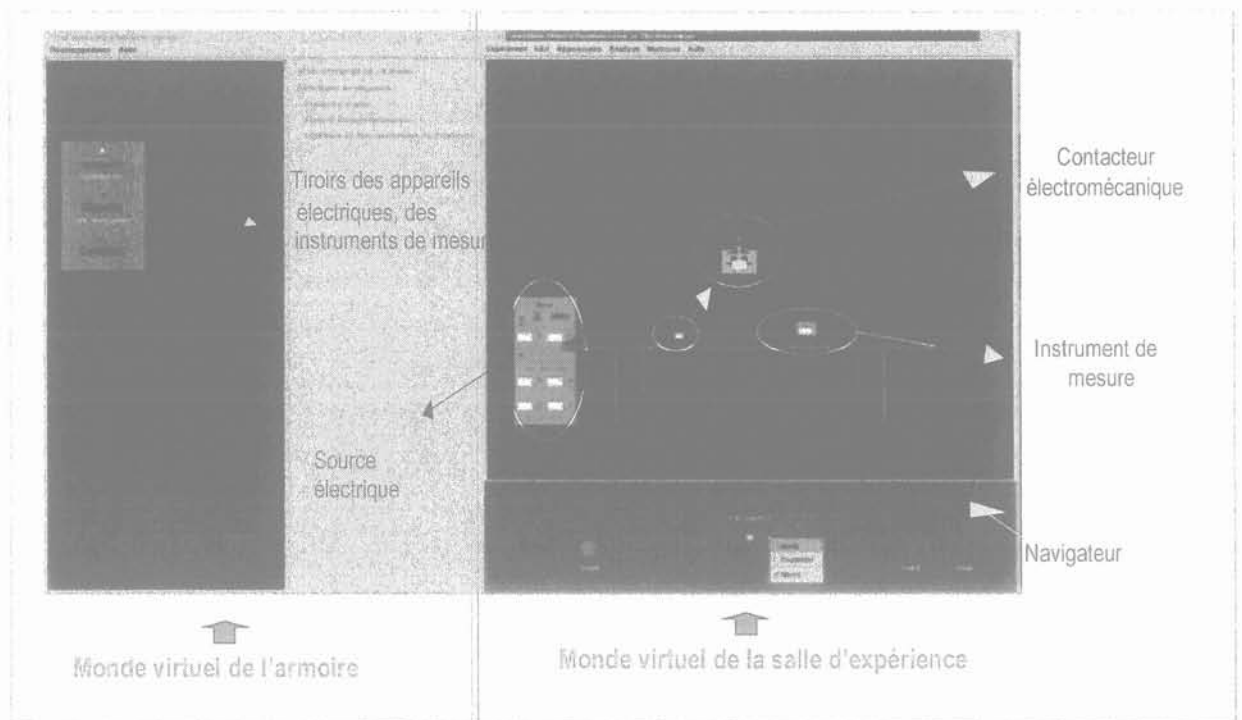


Figure V - 4 Présentation visuelle du LVEE

### 5.5.1. Monde virtuel de l'armoire

Dans ce monde virtuel, l'utilisateur interagit avec les "tiroirs" pour développer une manipulation d'expérimentation, en choisissant le dispositif électrique examiné, l'objectif d'expérimentation et les conditions d'expérimentation, au cas où l'objectif d'expérimentation choisi peut être atteint par plusieurs états de fonctionnement du dispositif (régime transitoire, régime permanent, à vide, ...). Les choix qui sont visualisés dans le tiroir des conditions d'expérimentation dépendent de la norme d'expérimentation correspondant à l'objectif d'expérimentation concerné.



La manipulation d'expérimentation ainsi développée se compose du dispositif électrique avec ses attributs, de la source électrique, et des instruments de mesure. Elle est visualisée dans le monde virtuel de la salle d'expérience.

### **5.5.2. Monde virtuel de la salle d'expérimentation**

Dans la salle d'expérience, se trouve une table sur laquelle est posée la manipulation d'expérimentation. L'expérimentation est effectuée dans cette salle. Le déplacement dans ce monde virtuel se fait à l'aide d'un navigateur. Nous utilisons deux types de navigation :

- *Examine;*

Ce type de navigation permet à l'utilisateur de visualiser la salle, au moyen de rotations, de translations et de zooms, comme si la salle était un objet tenu dans ses mains.

- *Walking;*

Cette navigation permet à l'utilisateur de marcher, de se déplacer et de se retourner dans la salle.

## **5.6. Interaction**

Les interactions entre l'utilisateur et les objets, dans les mondes virtuels du LVEE, correspondent aux étapes du processus d'expérimentation (§ table III-1). Comme c'est décrit sur la figure V-5, les interactions dans le monde virtuel de l'armoire représentent les activités dans les première et deuxième étapes du processus d'expérimentation. Ensuite, les interactions dans le monde virtuel de la salle d'expérience s'accordent avec les étapes de la mise en œuvre de l'expérimentation, de l'observation et de l'analyse.

Dans la salle d'expérience, l'utilisateur doit, tout d'abord, définir les valeurs des paramètres du dispositif examiné. Il s'agit, dans l'ordre, des caractéristiques de fonctionnement du dispositif et, pour chacun de ses composants, de la géométrie, du matériau et des contraintes, comme par exemple la raideur du ressort de rappel d'un contacteur. Ces valeurs seront utilisées pour dessiner la géométrie et les propriétés physiques du dispositif et, par des logiciels de CAO, pour en simuler le fonctionnement.

Ensuite, l'utilisateur doit mettre en place l'alimentation du dispositif, soit la tension, soit le courant de la source électrique. Dès que le dispositif est connecté à la source électrique, la méthode numérique qui sera utilisée pour calculer le fonctionnement du dispositif est demandée à l'utilisateur. Le choix de la méthode numérique implique le choix du logiciel de simulation. Ce choix dépend de l'objectif d'expérimentation donc du modèle physique du dispositif ou du domaine physique étudié.

Enfin, l'utilisateur peut observer les comportements physiques du dispositif qui sont visualisés sur les instruments de mesure et les animations du dispositif lui-même. Le calcul de ces comportements physiques s'effectue par la facette *Abstraction* de l'agent du montage d'expérimentation en utilisant les résultats de la simulation numérique faite par le logiciel de CAO. Les activités que l'utilisateur a la



possibilité de réaliser dans la salle d'expérience sont affichées dans une fenêtre du protocole d'expérimentation qui apparaît en même temps que l'affichage de la manipulation d'expérimentation.

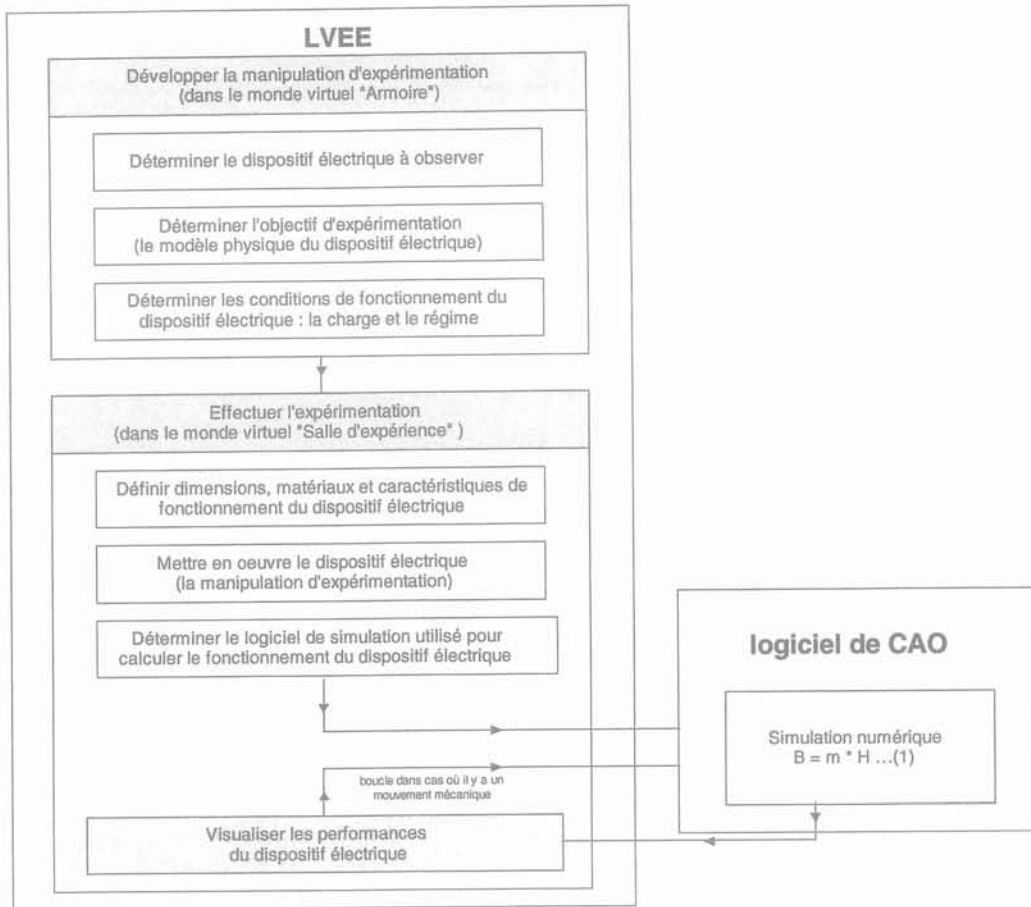


Figure V - 5 Processus d'expérimentation dans le LVEE

## 5.7. Pilotage des logiciels de CAO

Le pilotage des logiciels de CAO est réalisé par l'intermédiaire des outils de communication de ces logiciels et des interfaces entre ces outils de communication et le LVEE (figure V-6). Parmi les agents interactifs du LVEE, ce pilotage est contrôlé par l'agent "VELaboratory". L'annexe H décrit les composants de pilotage du logiciel Flux2D et celui de l'application de calcul ElementsRoters (§ annexe K).

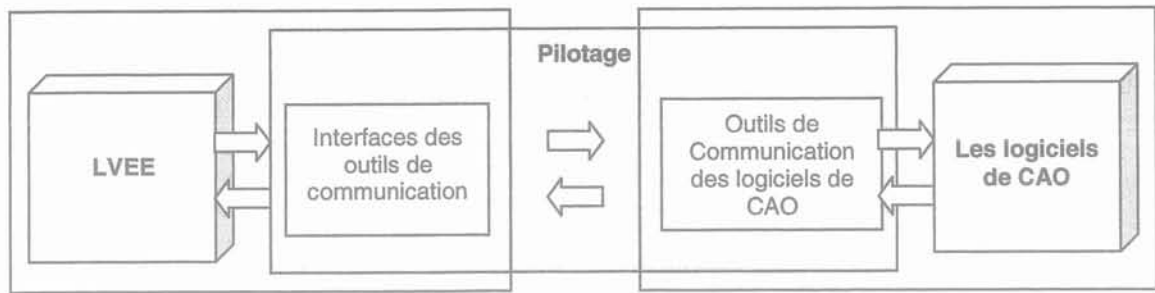


Figure V - 6 Structure de pilotage des logiciels de CAO

### 5.7.1. Outils de communication

En principe, l'outil de communication d'un logiciel de CAO est développé pour que l'on puisse accéder à un scénario de simulation disponible et communiquer avec le code de simulation numérique. En accédant au scénario de la simulation, il modifie les valeurs des paramètres d'entrées. Ensuite, après avoir mis en œuvre le code de simulation, il récupère les nouveaux résultats de simulation numérique.

L'outil de communication pour le logiciel flux2D ainsi que celui pour l'application de calcul ElementsRoters, a été conçu en considérant que la communication peut se réaliser dès lors que l'on peut leur fournir des données (une géométrie, un paramétrage, des directives de maillage, ...) qui correspondent au modèle du dispositif électrique examiné dans le LVEE. Autrement dit, on doit au préalable créer, à l'intention de ces logiciels de simulation, des projets qui représentent tous les dispositifs et toutes les expérimentations qui seront disponibles dans le LVEE.

### 5.7.2. Interfaces des outils de communication

L'interface de l'outil de communication est chargée de la traduction des données produites lors des activités dans les mondes virtuels du LVEE pour qu'elles puissent être utilisées par l'outil de communication. Elle connaît donc à la fois la structure des données du LVEE et celle de l'outil de communication qui permet de définir les données du logiciel de CAO correspondant.

## 5.8. Classes des agents interactifs du LVEE

Les classes des agents interactifs du LVEE sont indépendantes les unes des autres. Elles sont liées par une relation d'association aux classes des composants AMF (Agent, Facette, Administrateurs, Port, ...).

Par rapport à la programmation orientée objet de l'API Java, il existe deux stratégies possibles pour réaliser l'architecture AMF [Poquet 98]:

1. Les objets de l'application héritent directement des classes Java d'objets AMF.
2. Les objets de l'application sont liés à ces objets par une relation d'association.

En considérant que la première solution peut entraîner des difficultés liées à l'absence d'héritage multiple en Java et à la sécurité du mécanisme de communication entre les objets AMF, c'est la deuxième solution qui est préférée. Cependant, quelques classes de l'application doivent aussi hériter de classes appartenant à l'API Java, par exemple des classes graphiques AWT (*Frame*, *Applet*, ...). Le langage Java n'autorise certes pas l'héritage multiple (une sous-classe ne possède qu'une seule classe mère), mais permet l'implantation de plusieurs Interfaces. Les objets AMF sont donc définis comme des Interfaces. Ainsi, une classe de l'application peut, par exemple, hériter de la classe *Frame* et implémenter des objets AMF.

En définissant les objets AMF comme des Interfaces (un ensemble de méthodes vides), on prend toutefois un risque au niveau du mécanisme de communication puisque le programmeur sera dans l'obligation de le coder dans les méthodes des Interfaces. Ce risque est particulièrement élevé, lors du développement d'une grande application où la communication est très complexe.

Concernant l'indépendance entre les classes des agents interactifs du LVEE, elle se justifie pour les raisons suivantes :

- Bien qu'ils aient, plus ou moins, la même structure, chacun des agents interactifs a, en effet, un comportement particulier.
- Cette indépendance facilite la modification locale à tous les niveaux, sans mettre en cause le fonctionnement global du système (§ 4.4).

Cependant, ce choix peut s'avérer assez lourd au niveau de la programmation, car il faut alors créer tous les paramètres et tous les attributs chaque fois que l'on développe une nouvelle classe pour un nouvel agent.

Pour conclure sur la réalisation informatique du LVEE, nous développons d'abord des classes représentant les objets AMF avant ceux définissant les agents interactifs du LVEE.

### 5.8.1. Composants AMF

Les objets AMF sont considérés comme le moteur de gestion d'interactions AMF car leur rôle est de gérer collectivement le mécanisme de communication, ce qui correspond aux appels des traitements suite aux activations des ports de sortie (§ 4.3.2.1).

L'annexe I décrit la structure et le modèle OMT (méthode d'analyse et de conception orientée objet) des classes des composants AMF. Ces classes sont développées en considérant qu'elles doivent, au minimum, réaliser le mécanisme de communication. Nous ne traiterons pas ici des fonctions nécessaires pour contrôler automatiquement les erreurs au cours de la création d'objet AMF et les événements lors du mécanisme de communication. Pour ce qui est des administrateurs de contrôle (§ 4.3.2.2), nous avons créé d'autres administrateurs comportementaux : Administrateur de *séquence cible* et

Administrateur *simple multi cible* afin de traiter l'activation des ports cibles respectivement de la manière suivante :

- Les ports cibles sont activés en séquence, à partir du port défini en premier lieu jusqu'au dernier, dès qu'un port source est actif.
- Le port cible est activé s'il porte un nom identique au nom du port source actif.

### 5.8.2. Agents interactifs du LVEE

Les objets de l'application, qui sont, dans ce cas, les classes des agents interactifs du LVEE, sont principalement chargés de fournir les données et d'effectuer les traitements sur celles-ci. La liaison d'association entre les objets AMF et ceux du LVEE découle des agents du LVEE et de leurs facettes qui contiennent, en effet, une partie pour les connexions et les communications, définie par le moteur AMF, et une partie application pour les créations et les traitements des données, comme cela est montré sur la figure V-7.

#### 5.8.2.1. Structure des classes

Les classes des agents du LVEE sont définies en suivant les primitives de l'application du modèle AMF (§ figure IV-7). La classe d'un agent interactif du LVEE se compose, en principe, d'une classe de son composant *Présentation* et d'une classe de son composant *Abstraction*. Par contre, son composant *Contrôle* est déjà inclus dans la partie de moteur AMF. Les classes des composants et des attributs d'un agent interactif sont groupées dans un paquet ayant le nom de l'agent.

Concernant le mécanisme de communication, au niveau de la facette, on développe les ports sources et cibles. Ensuite, au niveau de l'agent, ces ports sont liés par des administrateurs définis dans la méthode de *communication()*. Les ports *exportés* sont redéfinis au niveau de l'agent pour qu'ils puissent être connectés aux ports appartenant, soit à l'agent parent, soit aux agents fils. Les ports *exportés*, destinés aux agents fils, sont définis dans la méthode d'*exportableForChild()*. Les ports *exportés* qui parviennent de l'agent parent, sont définis dans la méthode d'*exportableFromParent (Agent owner)*.

La fonction particulière, appelée "*démon*", qui est associée à un port de communication se définit par une classe dans la classe de la facette à laquelle ce port appartient. En plus, de la même façon, le rôle de traduction d'un administrateur est défini comme une classe privée dans la classe de l'agent qui contient cet administrateur.

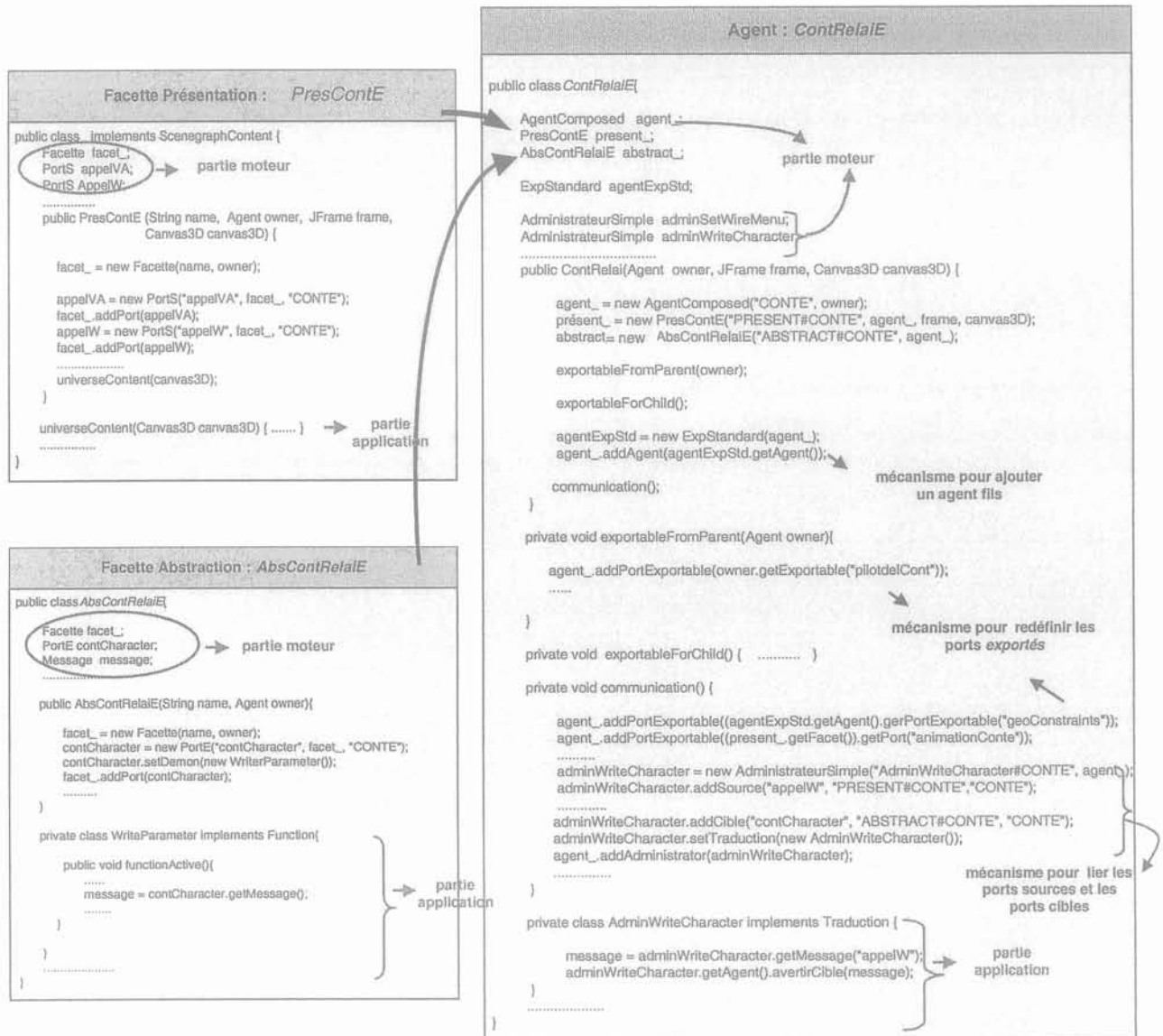


Figure V - 7 Classes représentant l'agent d'un contacteur avec l'électroaimant en forme de E

### 5.8.3. Extension des agents interactifs dans le LVEE

Pour cette première démarche, nous ne nous sommes concentrés que sur les expérimentations des contacteurs électromécaniques. L'extension des agents interactifs dans le LVEE sera donc nécessaire pour des agents représentant d'autres dispositifs électriques et leurs montages d'expérimentation.

Pour ajouter un nouvel objet (agent interactif de dispositif électrique ou de montage d'expérimentation) à la structure des classes des agents interactifs du LVEE, la marche à suivre est la suivante :

- Définir cet objet comme agent fils d'un des agents représentant les catégories des dispositifs électriques ou les catégories des objectifs d'expérimentation.
- Connecter les ports *exportés* aux agents correspondants.
- Lorsque le nouvel objet est un agent interactif de montage d'expérimentation, ajouter, à la base de données, la norme d'expérimentation concernée.
- Au cas où le nouvel objet ainsi ajouté ne ferait pas partie des choix dans le monde virtuel de l'armoire, créer le nouveau choix pour cet objet et le lier à un port de sortie. Ensuite, connecter le port de sortie au mécanisme de communication pour que le nouvel objet puisse apparaître dans la salle d'expérience lorsque l'utilisateur clique sur son nom dans le monde virtuel de l'armoire.

Cependant, l'extension des agents interactifs dans le LVEE est complexe au niveau de la connexion manuelle des ports *exportés*. Il sera donc nécessaire, dans le futur, d'avoir un outil permettant de lier automatiquement les ports *exportés* lors que l'on ajoute un nouvel agent interactif.

#### 5.8.4. Distribution des composants de présentation

Une forte dispersion de l'interface globale est considérée comme l'inconvénient majeur des modèles multi-agents (§ 4.2.2). Cependant, à l'aide du modèle de programmation "*scene graph*" du Java3D, nous pouvons facilement construire la présentation globale du LVEE en assemblant des *scene graphs* définis par le composant *Présentation* de chaque agent.

La présentation visuelle globale du LVEE peut être décomposée en suivant la structure hiérarchique des agents interactifs (figure V-8). En définitive, la perception de cette présentation globale n'est plus difficile. De plus, grâce aux classes *Group* et en particulier la classe *BranchGroup*, on peut facilement attacher la présentation d'un agent à la présentation globale ou la détacher.

La figure V-9 présente le *scene graph* qui est défini par le composant *Présentation* de l'agent "*Contacteur*". Il groupe les présentations de l'agent "*ContRelaiE*", de l'agent "*ContRelaiU*" et de l'agent "*ContRelaiP*". Ensuite, la figure V-10 montre la présentation de l'agent "*ContRelaiE*" qui définit la géométrie et les comportements physiques du contacteur électromécanique avec l'électroaimant en forme de E. Le *scene graph* de cette présentation exprime la structure physique du contacteur. L'annexe J expose les *scene graphs* que nous proposons pour les transformateurs et les machines tournantes. Comme pour le contacteur, ils sont faits en suivant leur structure physique.

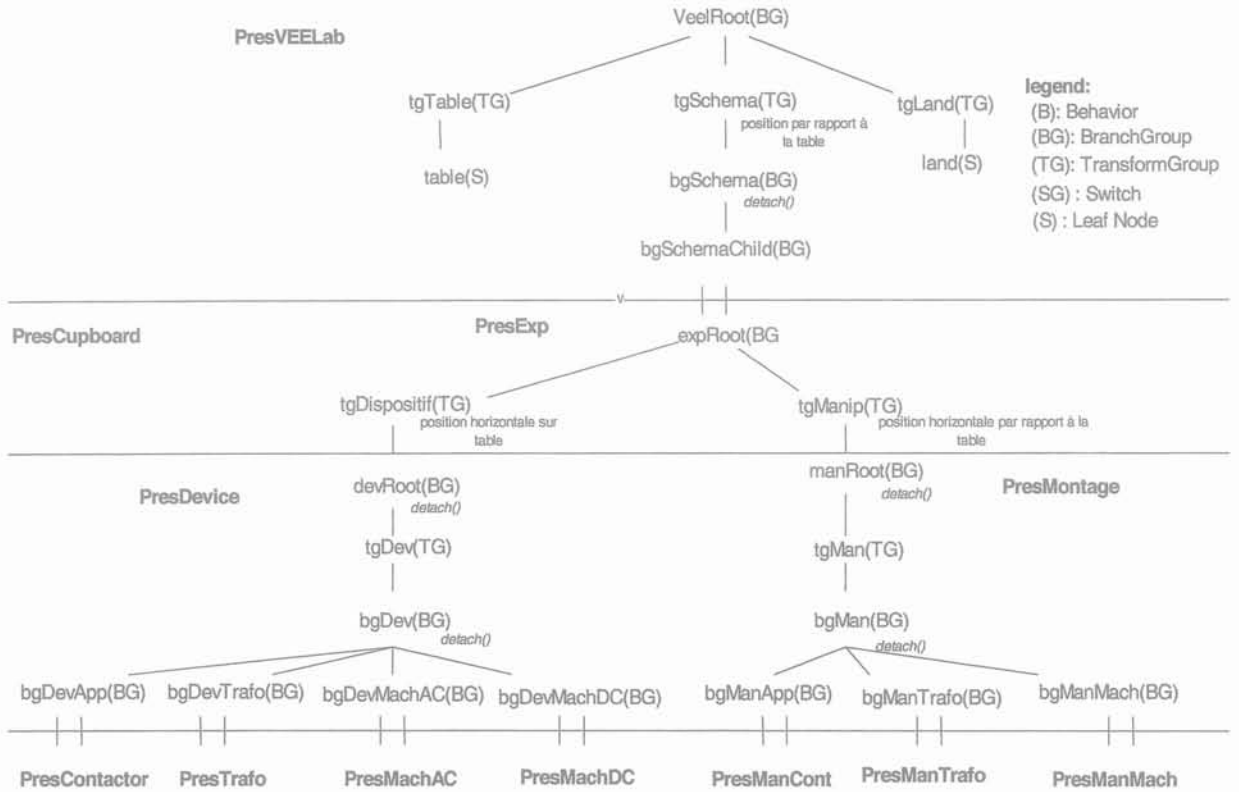


Figure V - 8 Scene graph des composants Présentation du LVEE

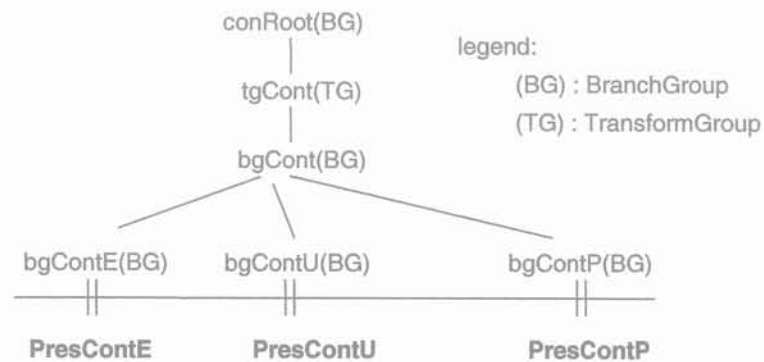
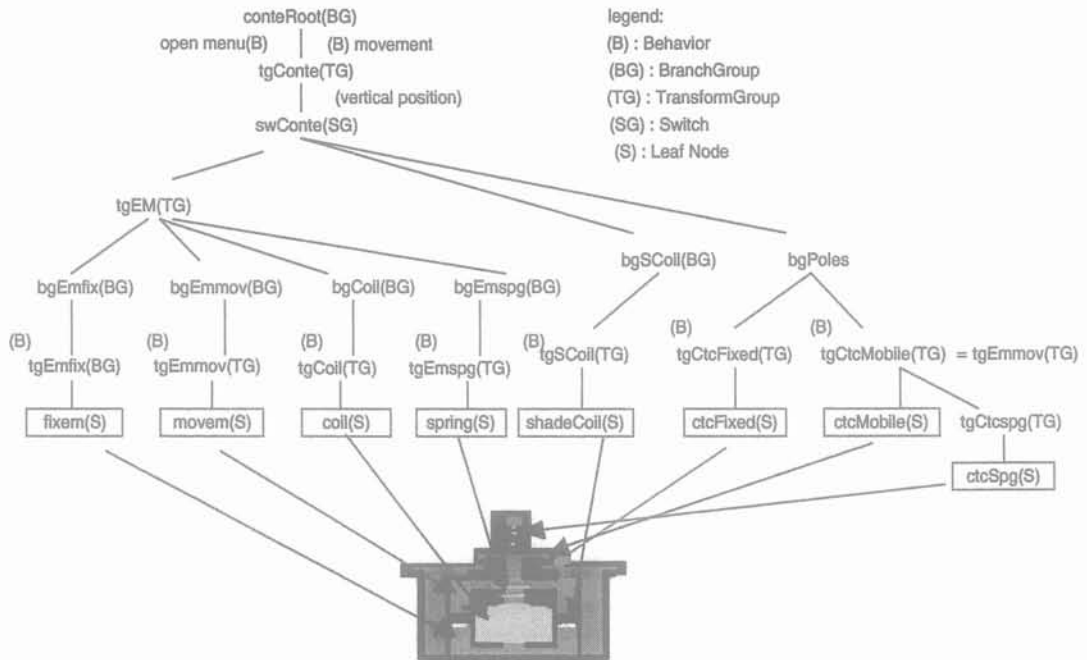
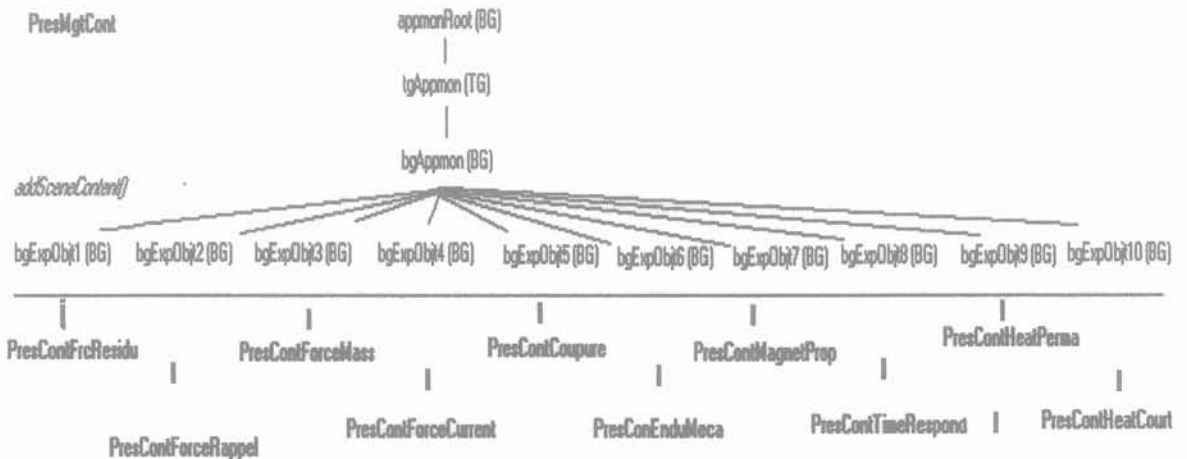


Figure V - 9 Scene graph du composant Présentation d'un contacteur

Figure V - 10 Scene graph du composant *Présentation* d'un contacteur en E

Le *scene graph* défini par le composant *Présentation* de l'agent "*ContacteurMontage*" est décrit sur la figure V-11. Ce *scene graph* est attaché à la présentation de l'agent "*Montage*" et groupe les présentations des agents représentant les montages d'expérimentation pour tous les objectifs d'expérimentation des contacteurs. La figure V-12 présente la présentation de l'agent "*ContacteurForceResiduMontage*" qui est le montage d'expérimentation utilisé pour étudier la force résiduelle se produisant dans un contacteur.

Figure V - 11 Scene graph du composant *Présentation* du montage d'expérimentation pour des contacteurs



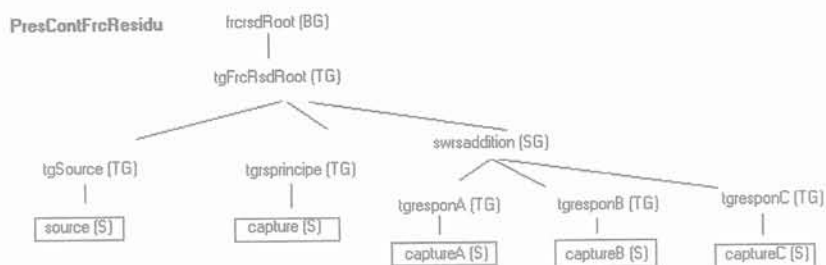


Figure V - 12 *Scene graph* du composant *Présentation* de montage pour étudier la force résiduelle d'un contacteur

Par contre, le *scene graph* défini par le composant *Présentation* de l'agent "Cupboard" (figure V-13) n'est pas attaché à la présentation du monde virtuel de la salle d'expérience car il exprime la présentation visuelle du monde virtuel de l'armoire.

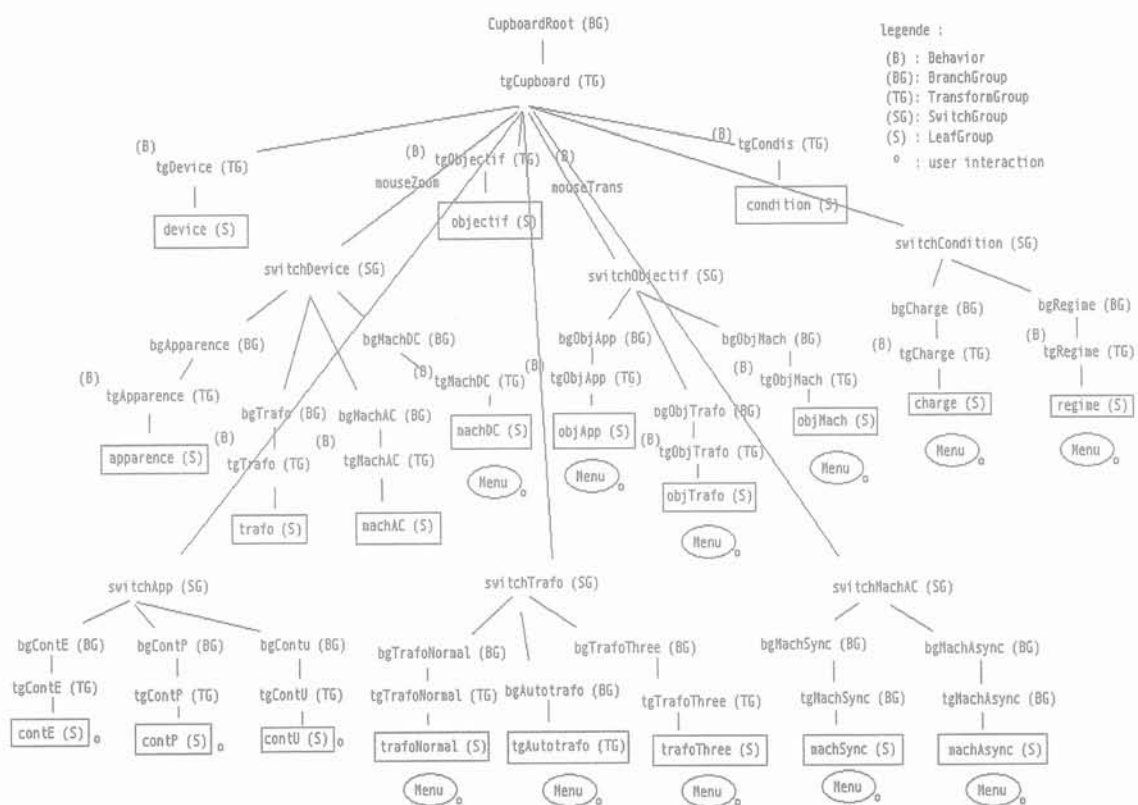


Figure V - 13 *Scene graph* du composant *Présentation* de l'agent "Cupboard"

## 5.9. Conclusion

Le Laboratoire Virtuel d'Expérimentation en Electrotechnique (LVEE) est un système interactif qui, au sein des logiciels de CAO, sert à la gestion des données. Il est chargé de créer les données nécessaires aux logiciels, de contrôler le processus de simulation et d'interpréter les résultats de simulation. Dans la mesure où il représente un Laboratoire d'Expérimentation en Electrotechnique, nous souhaitons que les activités dans le LVEE soient transférables comme expérimentations dans le laboratoire réel. De plus, les informations sont structurées pour qu'il soit facile de comprendre ce que représentent les données contenues dans le dispositif et de connaître leurs importances dans le fonctionnement du dispositif.

A l'aide du modèle de programmation "*scene graph*" de Java3D, nous pouvons aisément répartir la présentation visuelle du LVEE parmi les agents interactifs en suivant leur structure hiérarchique. De plus, ce modèle de programmation nous permet de définir un modèle de dispositif électrique qui s'inspire de sa structure physique.

L'apport de la technologie de Réalité Virtuelle a pour objectif d'avoir une visualisation naturelle des comportements physiques des dispositifs électriques. Cependant, l'implémentation des technologies avancées de l'interface de Réalité Virtuelle sera nécessaire pour que l'on puisse vraiment interagir avec les données dans le LVEE, c'est-à-dire la modification des paramètres du dispositif et l'observation naturelle de ses comportements physiques.

Bien sur, il existe encore beaucoup à faire pour le développement du LVEE. Voici quelques éléments à réaliser :

1. Le développement d'un outil qui permette de lier automatiquement les ports *exportés* au mécanisme de communication existant, car la connexion manuelle est assez lourde. En particulier, pour une grande application, le schéma du mécanisme de communication entre des agents interactifs devient complexe, comme on peut le vérifier dans l'annexe N pour le LVEE.
2. L'amélioration des fonctionnalités du pilotage des logiciels de CAO de manière, au travers des interactions dans le LVEE, à véritablement créer un projet de calcul sur les logiciels et à utiliser au mieux leur puissance pour calculer les comportements physiques des dispositifs électriques.
3. L'implémentation des technologies avancées d'interface de Réalité Virtuelle pour que le LVEE puisse servir comme on le souhaite.



## Chapitre 6

# **Application du Laboratoire d'Expérimentation en Electrotechnique**



## Chapitre 6

### Application du Laboratoire Virtuel d'Expérimentation en Electrotechnique (LVEE)

#### 6.1. Introduction

Ce chapitre est consacré à l'application du Laboratoire Virtuel d'Expérimentation en Electrotechnique (LVEE) pour l'expérimentation virtuelle de contacteurs électromécaniques. Nous allons utiliser le logiciel Flux2D industriel et le programme de calcul ElementsRoters, comme logiciels de simulation pour calculer le fonctionnement d'un contacteur. Flux2D est un logiciel de simulation qui met en œuvre la méthode des éléments finis, alors que ElementsRoters est un programme de calcul des grandeurs magnétiques basé sur le calcul de permeances par les éléments de Roters (§ annexe K).

A travers cette application, nous présenterons, pas à pas, les mécanismes de calcul et de visualisation des comportements physiques de dispositifs électriques dans le LVEE, ici un contacteur avec l'électroaimant en forme de E, par rapport à la simulation numérique faite par des logiciels de simulation. Nous montrerons que ces mécanismes sont réalisés par l'intermédiaire des composants des agents représentant le contacteur et le montage d'expérimentation. Enfin, nous terminerons ce chapitre en décrivant les impacts des utilisations du LVEE envisageables dans les domaines de l'éducation et de l'industrie.

#### 6.2. Un exemple : l'expérimentation de contacteur

La première chose à faire avant d'utiliser le LVEE est de définir le dispositif électrique à étudier et les objectifs à atteindre. Pour notre démonstration, nous choisissons donc un contacteur ayant un électroaimant en forme de E. Notre objectif d'expérimentation va être l'étude de la variation de la force magnétique de l'électroaimant par rapport au courant de la bobine.

Le dispositif à examiner et les objectifs d'expérimentation étant définis, nous pouvons nous déplacer dans la salle du LVEE afin de préparer la manipulation. En premier lieu, nous allumons la lumière dans le laboratoire en choisissant le sous-menu "Nouveau" dans le menu "Expérience" du menu barre situé au-

dessus de la fenêtre de la salle. Ensuite, nous interagissons avec les données du monde virtuel de l'armoire afin de déterminer la composition de la manipulation d'expérimentation :

- Parmi l'ensemble des tiroirs consacrés aux dispositifs, nous cliquons sur celui étiqueté *ContE*, pour sélectionner un contacteur possédant un électroaimant en forme de E. Le contacteur ainsi choisi apparaît sur la table de la salle d'expériences (figure VI-1).

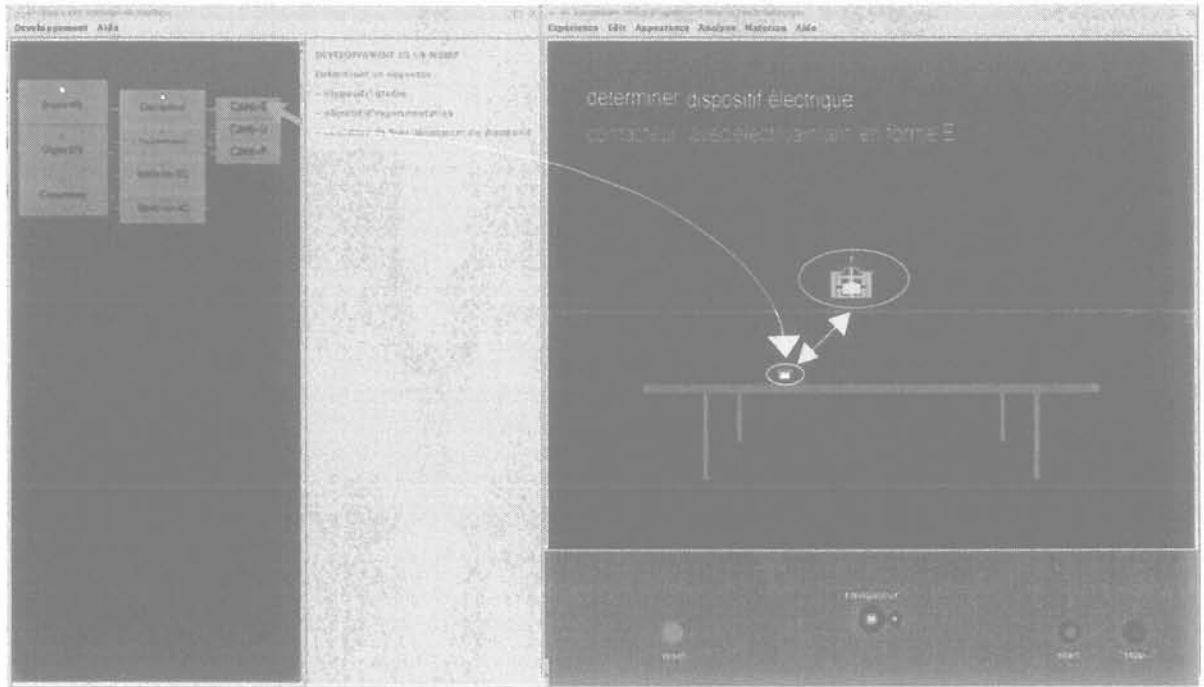


Figure VI - 1 Développement de la manipulation d'expérimentation : déterminer le dispositif examiné

- Dans le 'tiroir' des objectifs d'expérimentation, et plus précisément dans la partie réservée aux contacteurs, nous cliquons sur le choix "*mesure de force (courant de la bobine)*" qui correspond à l'objectif d'expérimentation que nous nous sommes fixé. Le montage d'expérimentation, et en particulier la source électrique et les instruments de mesure, nécessaires pour l'objectif d'expérimentation concerné, apparaissent alors dans la salle d'expérimentation (figure V-2). En même temps, un panneau contenant le protocole d'expérimentation s'affiche.

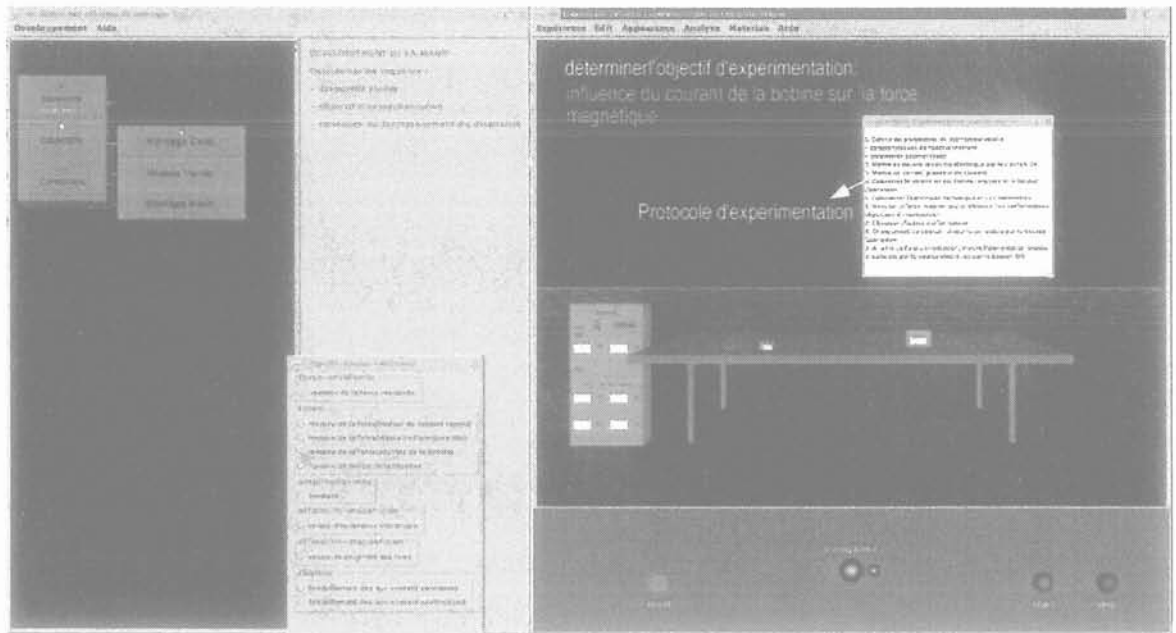


Figure VI - 2 Développement de la manipulation d'expérimentation : déterminer l'objectif d'expérimentation

- Dans le ' tiroir ' des conditions d'expérimentation, nous choisissons le régime dans lequel le contacteur doit fonctionner pour que l'objectif d'expérimentation puisse être réalisé. Il s'agit du régime mécanique (régime transitoire ou régime permanent) et du régime de la charge (à vide, en court circuit ou en charge). Pour l'objectif particulier de l'exemple choisi, nous n'avons qu'un seul choix qui est le régime 'transitoire mécanique' (figure VI-3) et pour la charge, il n'existe que le choix à vide, qui est appelé ici 'non chargé' (figure VI-4).

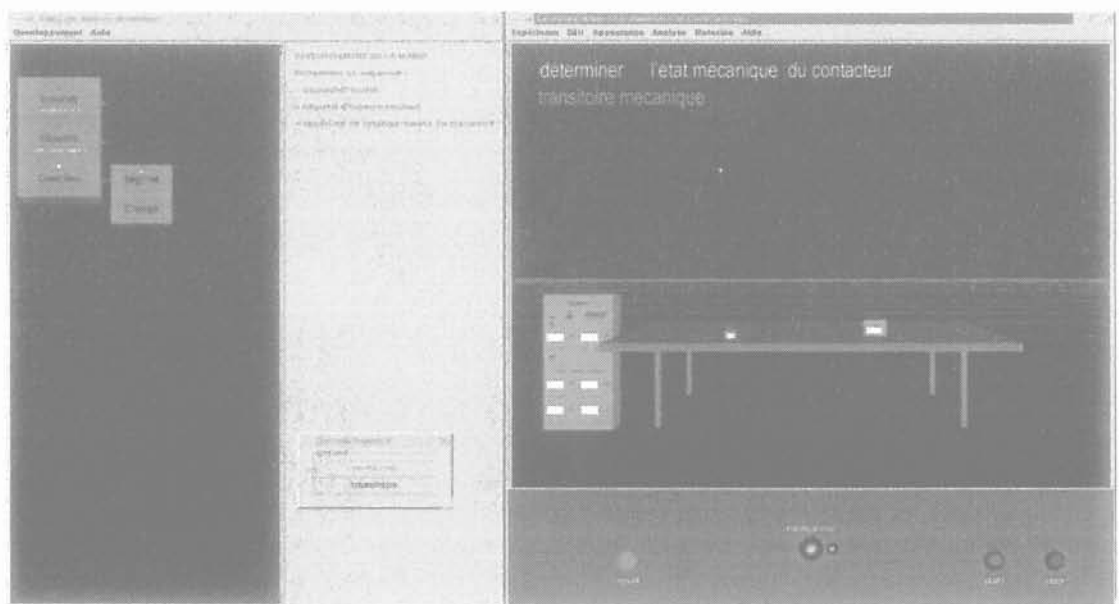


Figure VI - 3 Développement de la manipulation d'expérimentation : déterminer l'état mécanique du dispositif



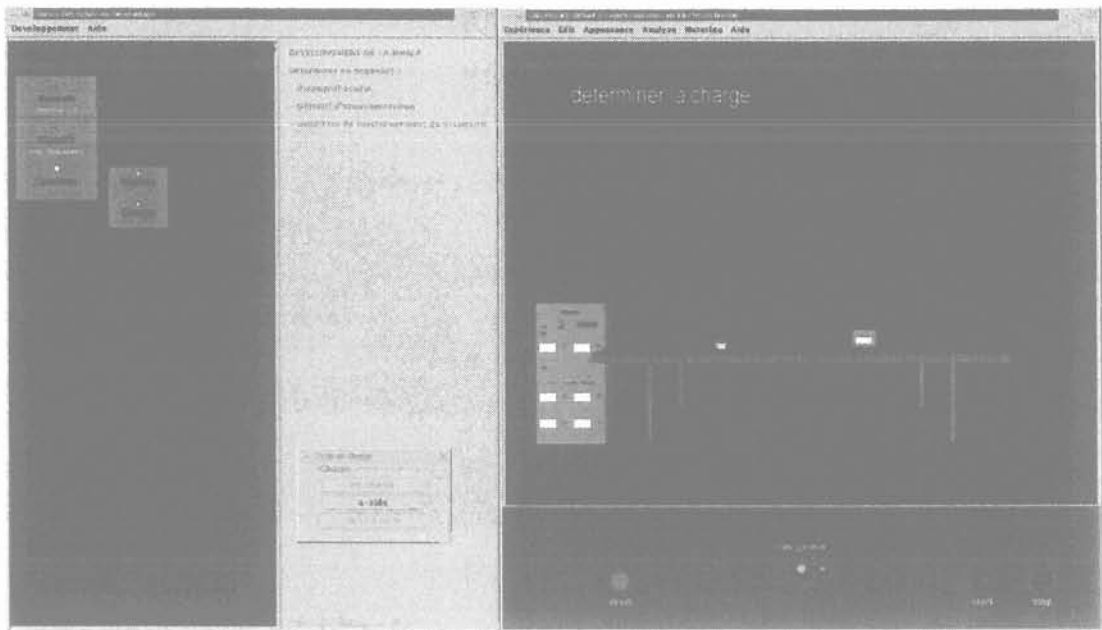


Figure VI - 4 Développement de la manipulation d'expérimentation : déterminer la charge du dispositif

Après la mise en place de la manipulation d'expérimentation, nous commençons à effectuer l'expérimentation.

#### - Définition des paramètres du contacteur

En cliquant (bouton gauche de la souris) sur le contacteur, un menu contextuel apparaît. Il contient la liste des types des paramètres que nous devons définir (figure V-5). L'ordre des types de paramètres dans cette liste correspond à leur ordre de définition. Pour le contacteur, il faut d'abord définir ses caractéristiques de fonctionnement, car elles auront des influences sur les contraintes de chaque composant. Par exemple, pour la bobine d'un contacteur à courant continu, il existe deux types de bobine envisageables : une bobine normale et une bobine à large plage. Les paramètres correspondant aux caractéristiques de fonctionnement du contacteur sont affichés dans un menu lorsque l'on clique sur l'item "caractéristiques".

Ensuite, nous allons aux paramètres de chaque composant. Dans le menu contextuel, les composants du contacteur sont présentés selon sa structure physique. Lorsque l'on clique sur l'item d'un composant, comme par exemple "bobine", on aura en premier un menu demandant les paramètres géométriques. Par la suite apparaît le menu pour définir les matériaux et leurs caractéristiques. Le menu suivant sert à déterminer les contraintes sur la bobine.

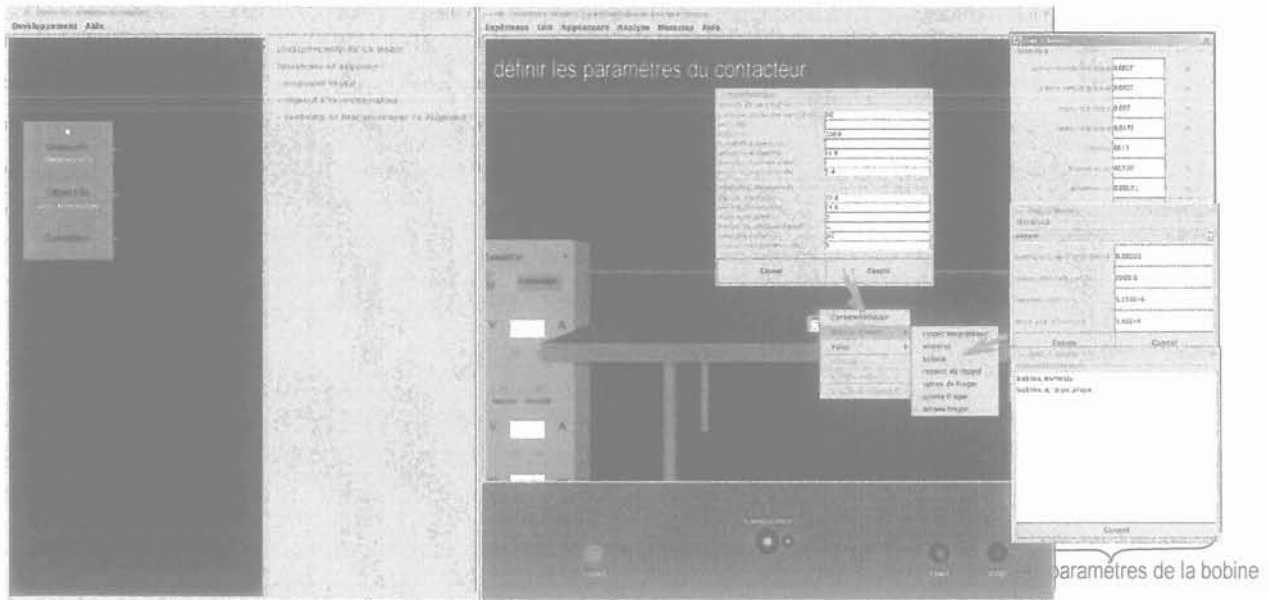


Figure VI - 5 Mise en œuvre l'expérimentation : définir les paramètres du contacteur

L'ordre de définition de chaque composant est important car il permet de réaliser les contrôles géométriques entre les composants. Si l'on suit cet ordre, on peut avoir un message d'avertissement, par exemple quand la section de la bobine nécessite une surface plus grande que la surface disponible dans le circuit magnétique (figure VI-6). Les valeurs des paramètres du contacteur que nous examinons sont décrites dans le sous chapitre 6.2.1.

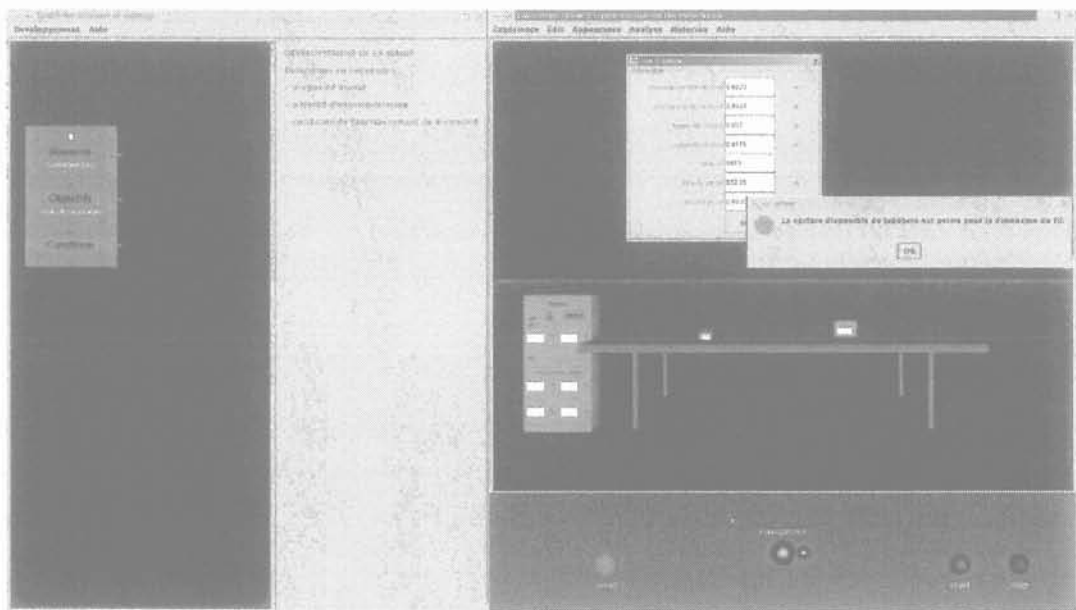


Figure VI - 6 Mise en œuvre l'expérimentation : message concernant la relation géométrique entre la bobine et le circuit magnétique du contacteur.

Comme nous l'avons décrit dans le chapitre 3 (§ 3.4), la topologie du contacteur électrique est fixée. Ainsi, dans le menu des paramètres géométriques, on ne peut que modifier les dimensions. Par ailleurs, à l'aide du menu "Matériau", nous pouvons accéder aux matériaux de la base des données et à leurs paramètres et créer de nouveaux matériaux (figure VI-7).

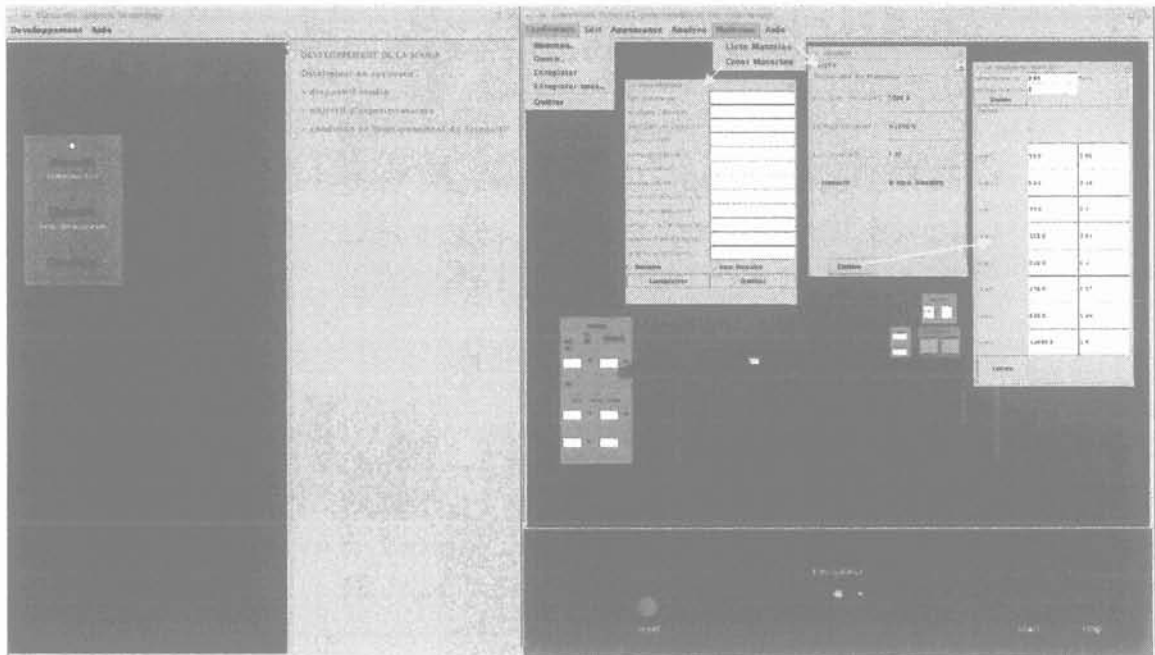


Figure VI - 7 Menu "Matériaux ".

- Faire fonctionner le contacteur

Après avoir défini les paramètres du contacteur, nous allons le faire fonctionner. Tout d'abord, nous activons les sources électriques en cliquant sur le bouton "On". Ensuite nous ajustons l'alimentation en utilisant les deux boutons situés au-dessous du cadran indiquant la valeur de la tension électrique. Le bouton de gauche permet de diminuer la valeur, tandis que celui de droite permet de l'augmenter. Lorsque le contacteur a été défini par l'expérimentateur comme un contacteur à courant continu, sur l'armoire des sources électriques, on ne peut que modifier la tension du courant continu. A l'inverse, s'il a été défini comme un contacteur à courant alternatif, on ne peut que modifier la tension du courant alternatif. Dans le cas d'un courant alternatif, on doit aussi définir les valeurs de la fréquence et du facteur de puissance. Ces valeurs sont obtenues de la plaque signalétique du contacteur. Lorsque le contacteur peut être alimenté, soit par la source de courant continu, soit par celle de courant alternatif, il faut préciser le type d'alimentation.

Ensuite, nous connectons la source électrique au contacteur en cliquant sur le bouton "Connexion". Cette connexion a pour conséquence de faire apparaître un menu, spécifique du monde virtuel, qui permet de choisir la méthode numérique souhaitée pour simuler le fonctionnement du contacteur. En fait,

cela correspond au choix du logiciel de CAO qui effectuera le calcul. Ces activités sont présentées sur la figure VI-8.

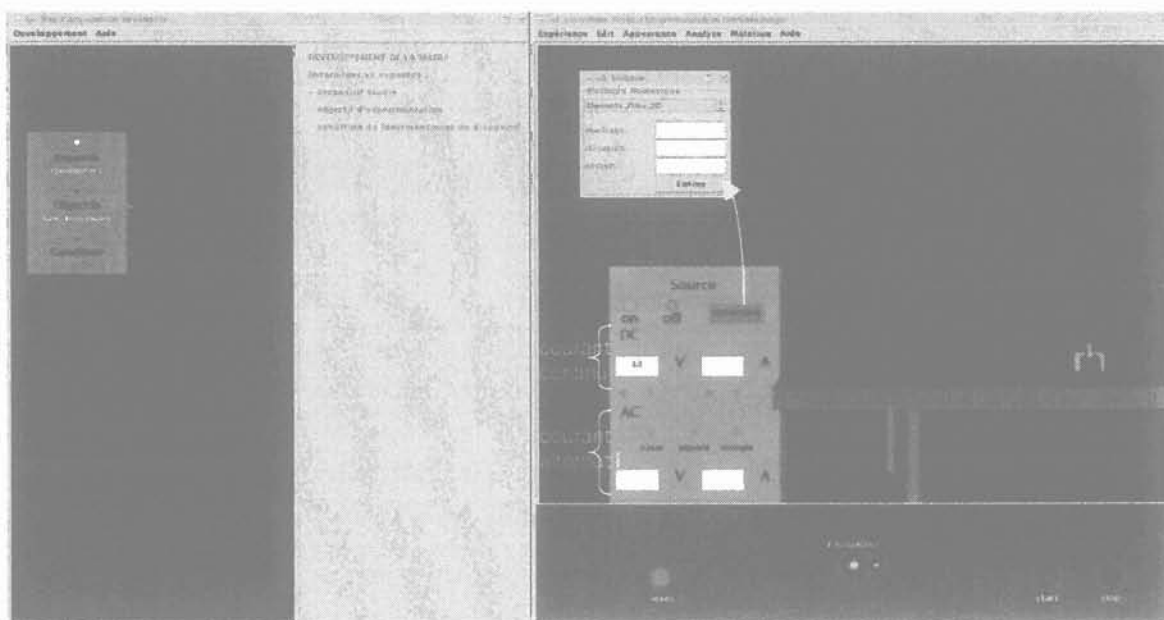


Figure VI - 8 Mise en œuvre l'expérimentation : faire fonctionner le contacteur

#### - Observation

Une fois que le processus de simulation numérique est fini, la force magnétique, qui est la réponse principale de l'objectif d'expérimentation, est affichée sur l'instrument de mesure en place (figure VI-9). Sur le contacteur lui-même, on peut constater le déplacement de l'armature mobile, si la force magnétique est suffisante pour le déplacer. Lorsque le contacteur est dans une phase d'ouverture, on peut aussi voir le phénomène d'arc électrique. Tous ces comportements physiques se prêtent bien à une visualisation naturelle.

A part la force magnétique, on peut observer, au travers d'instruments de mesure, d'autres comportements physiques, comme l'induction magnétique, le champ magnétique, les flux magnétiques et l'effet des spires de Frager (dans le cas d'un contacteur à courant alternatif). Tous les comportements accessibles sont listés dans un menu qui apparaît en même temps que l'animation des comportements physiques sur le contacteur et la visualisation de valeur numérique de la force magnétique sur l'instrument de mesure ou bien, à la fin du processus de simulation numérique. Dans ce menu, un bouton "courbes" permet aussi de visualiser des courbes exprimant la relation entre ces comportements physiques. Les résultats pour chaque configuration de valeurs des paramètres du contacteur sont présentés dans le sous chapitre 6.2.3.

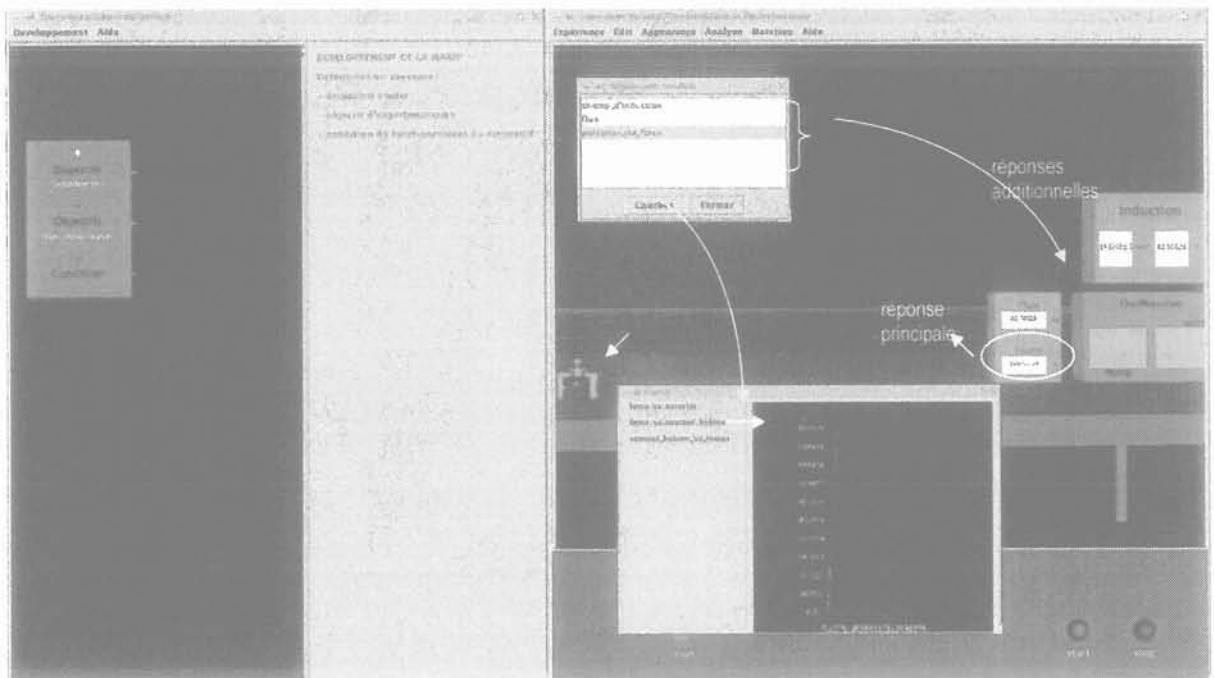


Figure VI - 9 Observation

#### - Analyse

Pour faciliter l'analyse des résultats de cette expérimentation, nous pouvons accéder aussi aux informations théoriques qui sont disponibles dans la norme d'expérimentation. Elles sont obtenues en cliquant le menu "Analyse" sur le menu barre de la fenêtre de la salle d'expériences (figure VI-10).

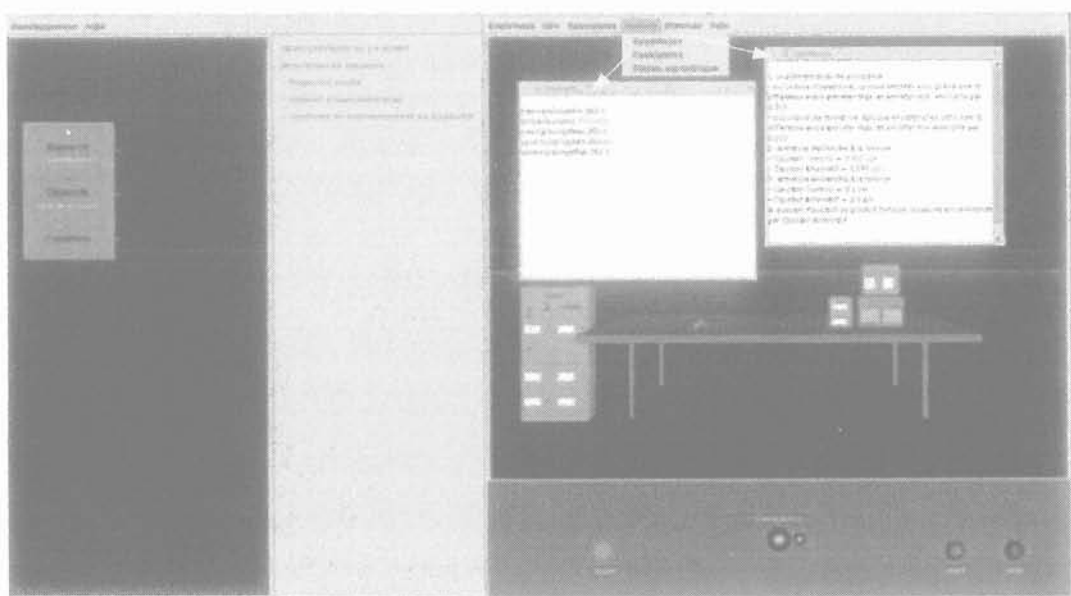


Figure VI - 10 Analyse

Lorsque l'on veut examiner une autre configuration de valeurs des paramètres du contacteur, il faut d'abord abaisser la source électrique jusqu'à zéro puis l'éteindre en cliquant sur le bouton "Off" et fermer la fenêtre des réponses additionnelles. On peut alors reprendre toutes les étapes pour effectuer une nouvelle expérimentation en introduisant les nouvelles valeurs.

Il est possible de sauvegarder une expérimentation terminée en choisissant "Enregistrer sous" ou "Enregistrer" sous le menu "Expérience". Cette action conserve la manipulation d'expérimentation ainsi développée pour une éventuelle future réutilisation. Cependant, avant d'enregistrer ou même de quitter sans enregistrer, il est obligatoire de remettre à zéro toutes les valeurs des sources électriques et de les éteindre. Cette obligation est faite pour apprendre un comportement qu'il est impératif de respecter à la fin d'une vraie expérimentation dans un laboratoire réel.

Si l'on veut développer une autre expérimentation, il faut aller dans le menu "Expérience" et choisir "Nouveau" pour supprimer le montage d'expérimentation courant. On peut alors commencer à développer une nouvelle manipulation d'expérimentation.

Toutes les étapes présentées ci-dessus sont développées dans l'annexe O concernant les directives d'utilisation du LVEE.

### 6.2.1. Les valeurs des paramètres du contacteur

Le contacteur en forme de E que nous examinons peut avoir l'une des deux caractéristiques suivantes :

- Contacteur à courant continu avec matériau magnétique linéaire (caractère A).
- Contacteur à courant continu avec matériau magnétique non linéaire (caractère B).

Pour chaque caractère, nous observons les comportements physiques du contacteur à l'ouverture et à la fermeture. Les valeurs des paramètres du contacteur pour chaque caractère sont décrites sur la table VI-1. La signification des paramètres géométriques est présentée sur la figure VI-11 ci-dessous :

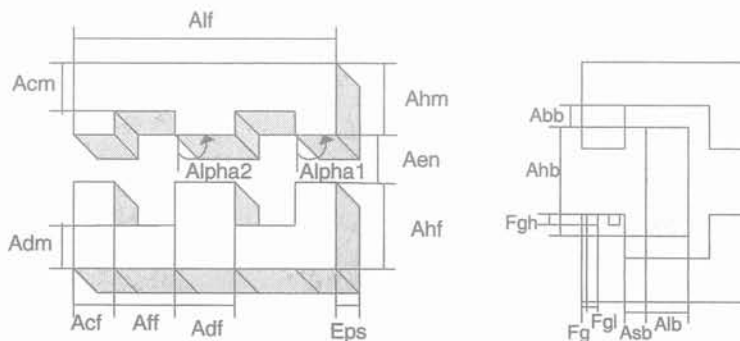


Figure VI - 11 Géométrie de l'électroaimant du contacteur examiné

Tableau VI - 1 La configuration des valeurs des paramètres du contacteur

Paramètres du contacteur					
Type		Nom	Valeur		
			Caractère A	Caractère B	
Caractéristiques de fonctionnement		categorie_d'alimentation (AC/DC) :	DC	DC	
		(AC)Un (V) :	-	-	
		(DC)Un (V) :	110	110	
		puissance_a_appel (VA) :	-	-	
		puissance_a_appel (W) :	9	9	
		puissance_au_maintien (VA):	-	-	
		puissance_au_maintien (W) :	9	9	
		courant_nominal (A) :	12	Idem	
		courant_thermique (A) :	25	Idem	
		nombre_de_poles :	3	Idem	
		nombre_de_poles_auxiliaires :	1	Idem	
		catégorie_d'emploi :	AC3	Idem	
		nombre_manœuvre (millions) :	2	Idem	
Géométrie (en m)	Circuit magnétique	hauteur de la branche mobile : (Ahm)		0.0177	Idem
		hauteur de la partie mobile : (Acm)		0.0047	Idem
		largeur de la branche : (Acf)		0.008	Idem
		distance entre branche : (Aff)		0.0116	Idem
		largeur de la branche central : (Adf)		0.0094	Idem
		Entrefer : (Aen)	ouverture	0.0002	Idem
			fermeture	0.006	Idem
		hauteur de la branche fixe : (Ahf)		0.0164	Idem
		hauteur de la partie fixe : (Adm)		0.0047	Idem
		épaisseur : (Eps)		0.0164	Idem
	Bobine	distance horisontale de circuit : (Asb)		0.002	Idem
		distance verticale de circuit : (Abb)		0.002	Idem
		largeur de la bobine : (Alb)		0.007	Idem
		hauteur de la bobine : (Ahb)		0.0176	Idem
		coil-tours :		6613	Idem
		longueur de coil :		11.675	Idem
		diametre de coil :		0.00012	Idem
	Spires de frager	distance de coté du circuit magnétique: (Fg)		0.0012	Idem
		largeur de surface du frager : (Fgl)		0.001	Idem
		hauteur de surface du frager : (Fgh)		0.002	Idem
		coil-tours :		1	Idem
		largeur de coil :		0.044	Idem
		diamètre de coil :		0.002	Idem
	Contacts	rayon de contacts :		0.002	Idem
		épaisseur de contacts :		0.001	Idem
	Matériau	Circuit magnétique	nom de matériau :	Fevlin	Fevsplin
		Entrefer		Vide	Idem



Paramètres du contacteur				
Type	Nom	Valeur		
		Caractère A	Caractère B	
	Bobine	Copper	Idem	
	Spires de frager	Aluminum	Idem	
	Contacts	Silver	Idem	
	Ressort de rappel	Platinum	Idem	
	Ressort de pôles	Platinum	Idem	
Contraints	Circuit magnétique	course totale de l'armature (m) :	0.006	Idem
		entrefer minimum (m) :	0.0001	Idem
		couche d'électroaimant :	1	Idem
	Bobine	type de bobine :	normale	Idem
	Ressort de rappel	raideur (N/m):	311	Idem
	Pôles	écrasement des contacts(m) :	0.0045	Idem
	Ressort de pôles	raideur (N/m):	100	Idem
Source électrique	DC Volt :	ouverture	1 et 2	1 et 2
		fermeture	75 et 88	75 et 88

Les valeurs des paramètres d'entrée sont déterminées en se référant au contacteur Télémécanique LP1 D12, en particulier pour les paramètres électriques [Gentil 88] [Telemecanique 91].

### 6.2.2. Le calcul et la visualisation des comportements physiques

La visualisation des comportements physiques du contacteur est conçue pour qu'elle puisse être naturelle. L'agent du contacteur en forme de E doit donc posséder les comportements d'un contacteur réel. Il devra d'ailleurs en être de même pour tous les autres dispositifs électriques.

Lorsque les comportements physiques des dispositifs électriques varient dans le temps, l'effet de temps réel est important. Malheureusement, le processus de simulation numérique prend un temps qui, bien sûr, devient plus long lorsque la structure de dispositif et les phénomènes modélisés sont plus complexes. Nous avons donc séparé le processus de simulation numérique et le mécanisme de la visualisation des comportements physiques. Cependant ils sont liés par un automatisme : dès que le processus de simulation numérique est fini, le mécanisme de visualisation est lancé automatiquement. Le départ de la visualisation est signalé par l'affichage du menu des réponses additionnelles. Dès cet instant, on peut commencer à observer la façon dont le dispositif réagit, en respectant les durées réelles des phénomènes.



Les relations du mécanisme de visualisation, par rapport à la simulation numérique effectuée par un logiciel de CAO et par rapport aux agents du montage d'expérimentation dans le LVEE, sont décrites sur la figure VI-12 suivante :

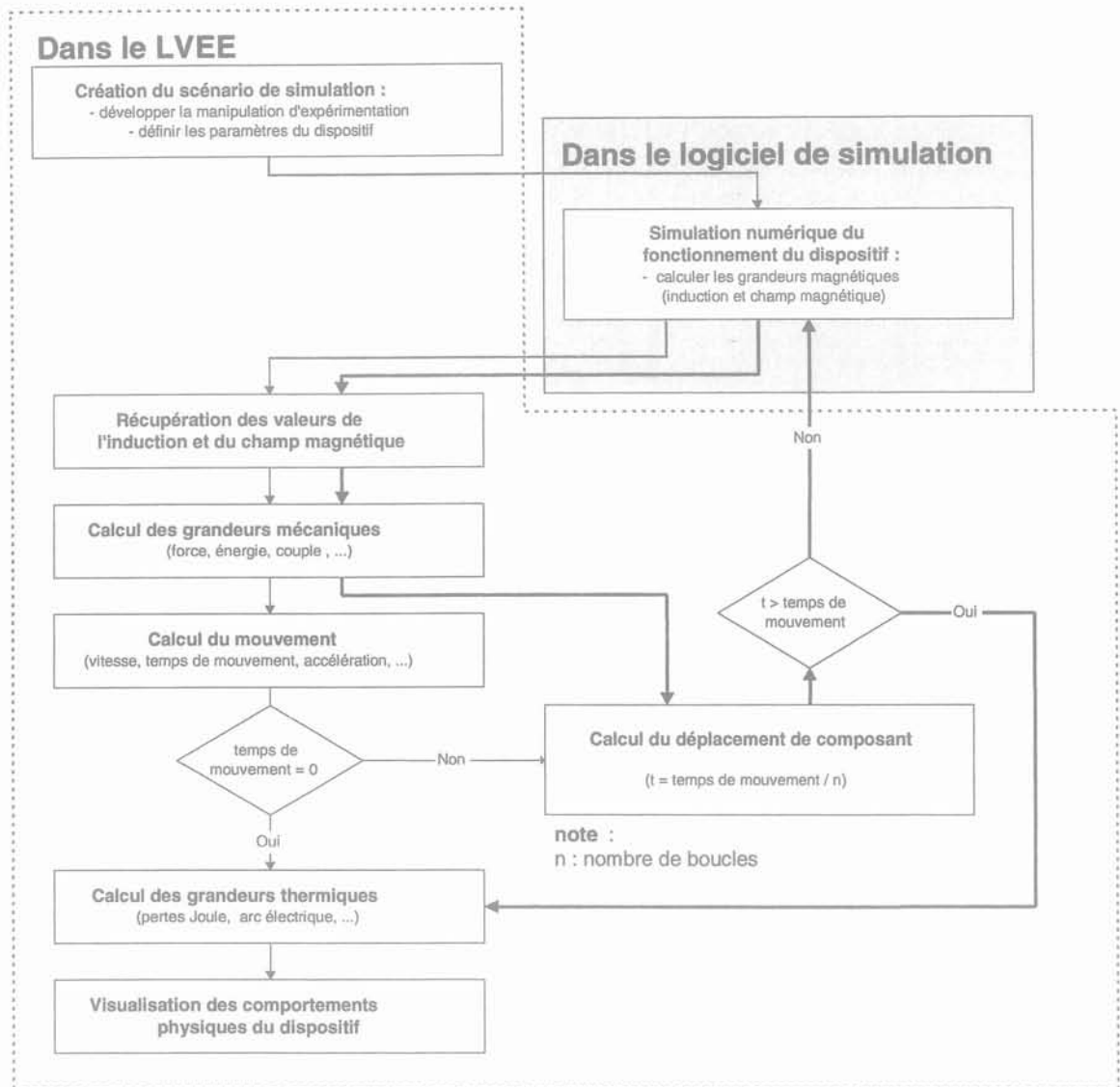


Figure VI - 12 Schéma du calcul et de la visualisation des comportements physique du dispositif

Le calcul des grandeurs magnétiques est refait à chaque changement de la géométrie concernant le champ magnétique, c'est-à-dire pour le cas du contacteur, à chaque déplacement de l'armature mobile. Lorsque l'agent de certain montage d'expérimentation découvre, à partir des résultats des logiciels de simulation, que des grandeurs mécaniques sont suffisantes pour produire un changement de la géométrie, il active une fonction de boucle afin de recalculer les grandeurs magnétiques et les grandeurs mécaniques de chaque position géométrique au cours du temps de déplacement. Le nombre de boucles dépend de la précision requise. Par ailleurs, les valeurs numériques affichées sur les instruments de

mesure sont les valeurs calculées lorsque les dispositifs arrivent à l'état permanent en terme mécanique aussi bien qu'électrique.

L'application du schéma présenté sur la figure VI-12 pour le contacteur électromécanique est décrite dans l'annexe P. Cette annexe contient aussi des explications sur la façon dont le composant *Présentation* et le composant *Abstraction* d'un agent interactif du montage d'expérimentation se chargent du calcul et de la visualisation, au travers des instruments de mesure, des comportements physiques du contacteur. Les performances physiques du contacteur sont visualisées par le composant *Présentation* de l'agent interactif le représentant.

Dans ce cas, la masse de l'armature mobile du contacteur est utilisée pour déterminer si la force magnétique peut engendrer un déplacement. Pour la simulation numérique sur le logiciel Flux2D, le maillage du contacteur a été développé en considérant qu'il doit rester valide pour toutes les valeurs de l'entrefer.

### 6.2.3. Les résultats


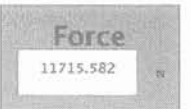
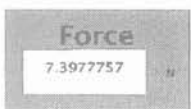


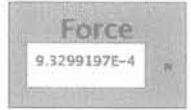

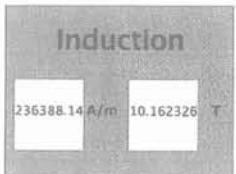

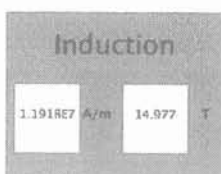

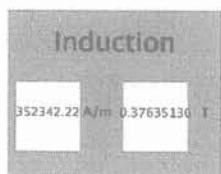
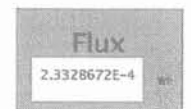
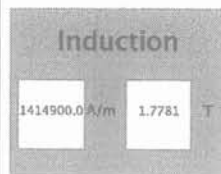

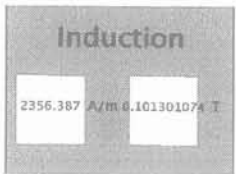
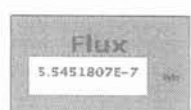
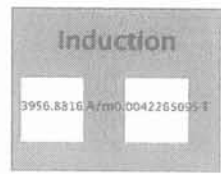
Comme nous l'avons décrit précédemment, les comportements physiques du contacteur que nous examinons, sont affichés, en terme numérique, sur les instruments de mesure et, en terme physique, par le contacteur lui-même. De plus, des courbes sont disponibles pour exprimer les relations entre ces comportements physiques. Les tables et les figures suivantes montrent les résultats de l'expérimentation concernant chaque configuration des valeurs de paramètres du contacteur.

#### 6.2.3.1. Numérique

La table VI – 2 présente les résultats visualisés à travers les instruments de mesure. Ils sont obtenus en alimentant le contacteur par une tension électrique de 1 volt pour l'ouverture et de 88 volts pour la fermeture.

Évidemment, les résultats sont sensibles aux hypothèses physiques ou aux logiciels de calcul utilisés pour évaluer le comportement physique. Par exemple en utilisant le logiciel Flux2D, on constate que la tension d'ouverture est inférieure à 1 volt, alors qu'avec le programme ElementsRoters, elle est égale à 1 volts pour le cas du matériau magnétique non linéaire et supérieure à 1 volts pour le cas du matériau magnétique linéaire.

Tableau VI - 2 Réponse en numérique

Réponse		Caractère			
		A		B	
		ElementsRoters	Flux2D	ElementsRoters	Flux2D
Principales	fermeture				
	ouverture				
Additionnelles	fermeture	 	 	 	 
	ouverture	 		 	

### 6.2.3.2. Graphique animé

La performance physique du contacteur se voit par le déplacement de l'armature mobile qui est solidaire des contacts. La figure VI – 13 montre le changement de position de l'armature. Lors de l'ouverture, on peut voir le phénomène d'arc électrique dans l'entrefer.

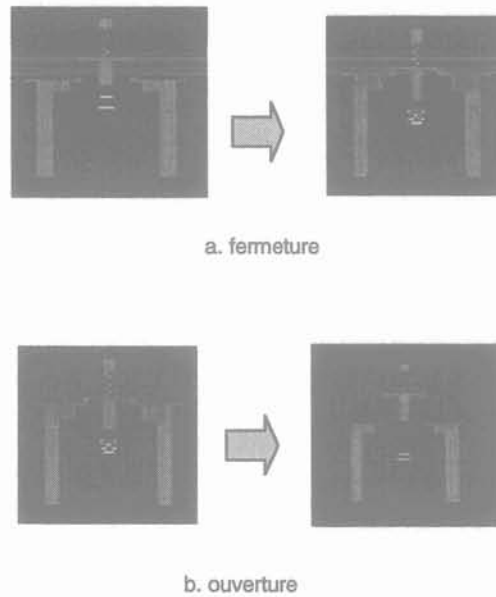
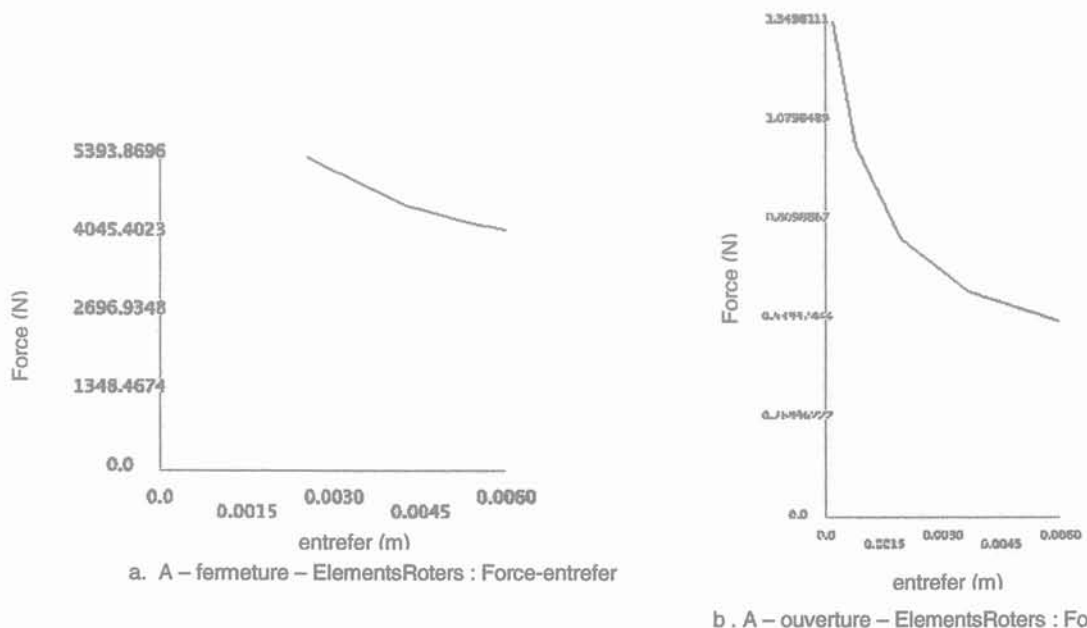


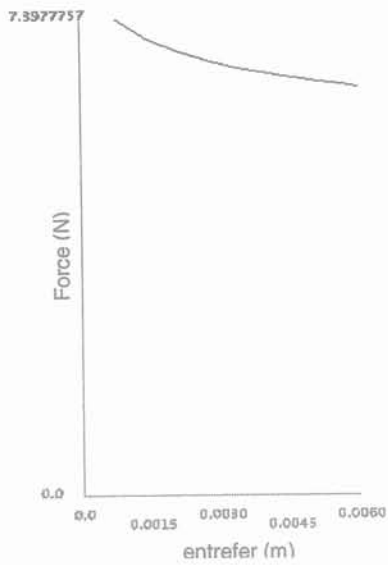
Figure VI - 13 Réponses physiques de contacteur

Nous vérifions que le contacteur se ferme plus vite lorsqu'il est alimenté par une tension électrique de 88 volts, qu'avec une alimentation de 75 volts. Nous constatons que la caractéristique du matériau a aussi de l'influence sur le temps de fermeture.

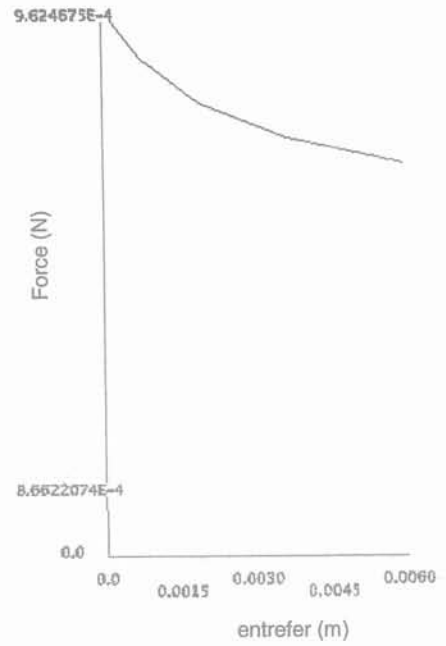
### 6.2.3.3. Courbes

Les relations à la fermeture et à l'ouverture "force – entrefer", et "force – courant de la bobine" sont présentées sur les figures suivantes :

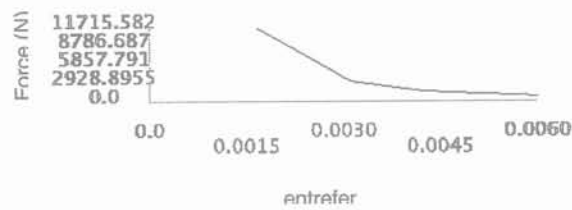




c. B – fermeture – ElementsRoters : Force-entrefer

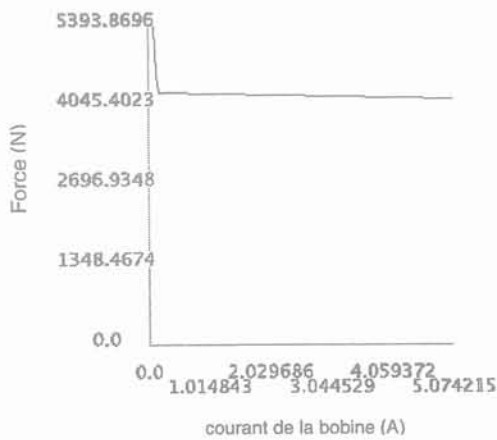


d. B – ouverture – ElementsRoters : Force-entrefer

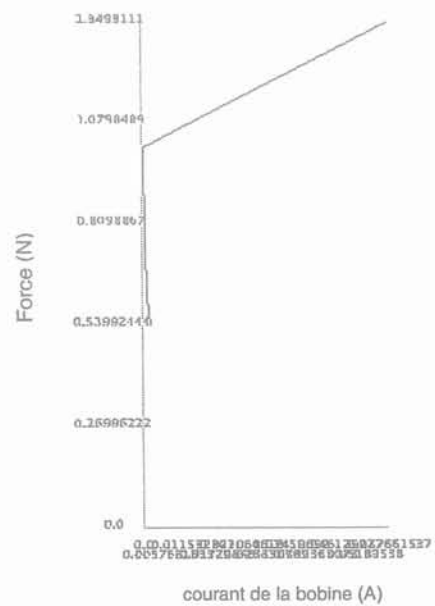


e. A – fermeture – Flux2D : Force-entrefer

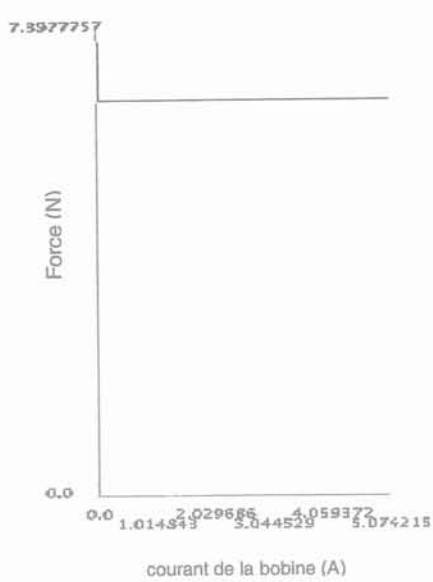
Figure VI - 14 Courbes : Force et entrefer



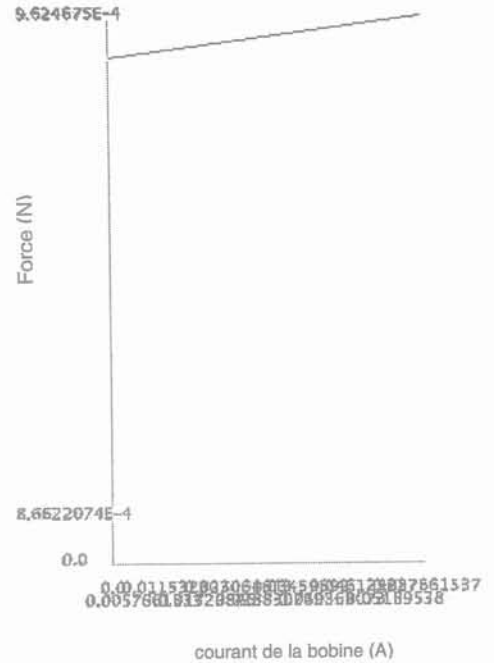
b. A – fermeture - ElementsRoters : Force-courant



a. A – ouverture - ElementsRoters : Force-courant



c. B – fermeture – ElementsRoters : Force-courant



certaines paramètres sur leur fonctionnement. Par ailleurs, au travers du LVEE, les étudiants auront une démarche expérimentale proche des standards du métier qui peut facilement s'appliquer aux expérimentations dans le laboratoire réel.

### **6.3.2. Industriel**

Pour l'industrie, on pense, en premier lieu, que le LVEE peut servir à la capitalisation du savoir-faire pour l'analyse du fonctionnement des dispositifs lors du diagnostic en cas de problème opérationnel. Plus tard, en implantant les technologies avancées d'interface de Réalité Virtuelle (§ 2.3) couplées à des simulateurs suffisamment rapides, nous pensons que le LVEE pourra vraiment devenir un système en mesure de remplacer les études chères ou/et dangereuses à réaliser dans le monde réel.

## **6.4. Conclusions**

Nous avons montré, en prenant le cas du contacteur, que le LVEE permet une exploitation plus naturelle des logiciels de CAO. Il facilite la compréhension des comportements physiques et centralise les informations et les paramètres nécessaires à toute la simulation. L'utilisateur se retrouve dans un environnement plus convivial et intuitif que l'environnement de calcul habituel.

L'état d'esprit "expérimentation", que nous trouvons dans le LVEE, nous conduit à une analyse logique des phénomènes physiques qui interviennent au cours du fonctionnement du contacteur. De plus, il devient plus facile de comprendre les influences de certains paramètres sur les comportements physiques étudiés. En particulier, l'avantage du LVEE, par rapport au laboratoire réel, est de pouvoir étudier plus facilement l'impact des changements de géométrie ou de matériau.

Cependant, la précision des résultats de simulation nécessite une attention particulière. En particulier, dans le cas d'une analyse par éléments finis, le maillage est essentiel lors des changements géométriques du modèle. Il est nécessaire que le maillage puisse s'adapter en fonction des dimensions et des phénomènes modélisés.

Nous avons développé l'application contacteur pour la validation du LVEE. Une deuxième application concernant les machines électriques tournantes et en particulier les moteurs asynchrones, est en cours de développement en collaboration avec EDF [Besnard 01]. Le but final pour EDF est de fournir à terme un outil intranet et convivial de centralisation des informations et de traitement des simulations pour les moteurs utilisés par EDF. Il devra en particulier faciliter la tâche des ingénieurs et techniciens en charge de la sélection et de la réception du matériel électrique, à qui l'on demande de plus en plus de remplacer les essais réels par des simulations. Le LVEE, paraît adapté à ce besoin dans la mesure où il permet d'obtenir de façon naturelle des informations sur les paramètres de moteur et sur son fonctionnement.

Bien entendu, en utilisant le clavier et la souris ordinaire, on ne peut qu'interagir avec les données à l'aide des menus. Ceci, malheureusement, amoindrit le sens de l'interaction intuitive. Plus que la visualisation graphique animée, nous souhaitons, à terme, pouvoir interagir directement avec les données et, même, ressentir certains comportements physiques des dispositifs qui ne sont pas visibles, tels que l'augmentation de température et l'existence de champ magnétique.

Le LVEE que nous avons réalisé n'est, en effet, qu'un prototype. Il ne possède pas encore toutes les fonctionnalités que nous avons introduites dans ses spécifications. Il reste encore beaucoup à faire pour qu'il puisse devenir un système robuste et convivial à souhait.





## **Conclusion générale**



## Conclusion générale

La conclusion de ce travail est organisée en deux parties. La première présente le bilan des propositions avancées dans les différents chapitres de ce mémoire. Nous consacrons la seconde partie aux perspectives que laisse entrevoir ce travail.

### I. Bilan des propositions

La proposition principale de ce travail est de considérer la démarche des techniques d'expérimentation des dispositifs électriques dans un laboratoire, comme une méthode d'exploitation des logiciels de CAO en Génie Electrique. L'objectif est de donner à ces logiciels, en particulier à ceux proposant l'analyse des comportements physiques des dispositifs électriques de façon intégrée, un accès proche des standards métiers, aussi naturel, et par conséquent aussi facile à comprendre que possible. L'approche des techniques d'expérimentation des dispositifs électriques contient les points positifs suivants :

1. L'utilisateur des logiciels de simulation ou le concepteur est conduit à développer librement ses idées en se focalisant sur le dispositif et non pas sur le simulateur.
2. Les paramètres du dispositif modélisé sont organisés selon sa structure physique, ce qui permet de percevoir plus facilement l'impact de leurs modifications sur le fonctionnement du dispositif.
3. Les travaux effectués sur logiciels de simulation correspondent à des activités qui seront exécutées dans le vrai laboratoire durant les tests du dispositif réel.

Cette approche a été réalisée dans un système interactif appelé "**Laboratoire Virtuel d'Expérimentation en Electrotechnique**" (LVEE). Nous avons anticipé, sans avoir vraiment pu l'implanter aussi complètement qu'il serait souhaitable, l'apport de la technologie de Réalité Virtuelle, afin d'avoir une interaction intuitive et une visualisation naturelle. Nous espérons qu'à l'aide de cette technologie, le LVEE pourra faciliter les techniques d'expérimentation visées et qu'il contribuera ainsi à la compréhension des phénomènes physiques qui interviennent dans le fonctionnement des dispositifs électriques. Cette compréhension dépend beaucoup de la qualité des modèles mis en œuvre dans les logiciels d'analyse. Cette qualité est précisément un des enjeux majeurs des logiciels de CAO en cours de développement pour lesquels le concept de LVEE devrait correspondre à une interface homme machine conviviale.

Le LVEE a été développé en modélisant tout d'abord un Laboratoire Réel d'Expérimentation en Electrotechnique. Le bilan que nous établissons pour le modèle ainsi développé est le suivant :

1. Les connaissances intervenant pendant une expérimentation de dispositif électrique correspondent bien aux informations nécessitées par les logiciels de CAO.
2. Les activités principales du processus d'expérimentation peuvent être assimilées aux activités de contrôle du processus de simulation.
3. Un Laboratoire Réel d'Expérimentation en Électrotechnique est un système contenant des objets indépendants qui interagissent fortement.

L'architecture multi-agents, AMF, que nous avons utilisée, a permis de respecter les indépendances entre les composants du laboratoire. Les points positifs de cette architecture pour le développement du LVEE sont :

1. Le mécanisme de référence indirecte, assuré par le composant *Contrôle*, permet de modifier ou de remplacer la réalisation d'une classe de compétences sans remettre en cause le fonctionnement des autres.
2. Les modifications logicielles pratiquées au niveau élémentaire sont applicables aussi à tous les niveaux d'abstraction. En particulier, il est aisé de remplacer la présentation d'un concept par d'autres objets interactifs ou d'ajuster localement l'image d'un objet élémentaire.

Comme avantages du LVEE actuel, qui est un système de Réalité Virtuelle sans immersion, nous voyons les points positifs suivants :

1. Par rapport à un outil de simulation seul, les informations ou paramètres nécessaires à toute la simulation sont centralisées et l'utilisateur se retrouve dans un environnement plus convivial et intuitif.
2. De plus, l'utilisateur bénéficie de l'apport des techniques d'expérimentation qui sont inhérentes au concept de laboratoire d'expérimentation.
3. Par rapport au laboratoire réel, il est plus facile d'étudier les influences de la géométrie et des caractéristiques de matériau.

Cependant, un certain nombre de limites apparaissent clairement dans notre réalisation :

1. Dans son état actuel, le LVEE ne contient que des expérimentations concernant les contacteurs.
2. L'interface graphique, non immersive, nécessite des améliorations.
3. Le lien avec les logiciels de simulation n'est capable que de piloter des simulations sur des modèles qui ont été préparés auparavant sur ces logiciels.
4. A l'aide de la souris 2D et du clavier, les sensations d'interactions avec les données sont limitées.
5. En tant que système interactif, il n'est pas encore robuste.

## II. Perspectives

Les perspectives d'amélioration du LVEE correspondent aux limitations que nous avons énumérées et sont donc les suivantes :

### 1. Extension du nombre de dispositifs connus du LVEE

- Il faudra étendre le nombre de dispositifs électriques traitables par le LVEE. Pour cela, il faudra accroître la population d'agents interactifs ce qui devrait être fait, non pas manuellement car c'est trop fastidieux, mais plutôt à l'aide d'un outil qui permette de créer automatiquement une instance de nouvel agent et de la lier au mécanisme de communication déjà disponible.

### 2. Implantation de la technologie d'interface de Réalité Virtuelle

- Un de nos objectifs principaux, avec la réalisation du LVEE, est de proposer une interface homme machine conviviale. Pour les étudiants, il s'agit de faciliter la compréhension des concepts théoriques de dispositif électrique. Pour les ingénieurs, il s'agit de mieux étudier et analyser les comportements physiques de dispositif électrique. Pour les scientifiques, il s'agit de visualiser les phénomènes physiques lors de l'étude de nouveau dispositif électrique. Pour vraiment atteindre cet objectif, il sera nécessaire d'implanter une technologie de Réalité Virtuelle plus performante que la souris 2D et le clavier. Il faudra être capable de mettre en œuvre une interaction avec les données plus intuitive et une observation des comportements physique de dispositif plus naturelle.

### 3. Intégration du LVEE dans les logiciels de CAO

- Il est aussi souhaitable que le LVEE puisse s'intégrer totalement avec les logiciels de CAO, non seulement pour la phase de simulation actuellement accessible, mais aussi pour la phase de description des modèles. Dans ces cas, le pilotage doit supporter la création de modèles par la manipulation d'objets. Il s'agit en quelque sorte d'adjoindre à la salle d'expérimentation virtuelle, un atelier virtuel ...



## **Bibliographie**





## Bibliographie

- [Barr 85] Barr, C. Paul, Roland L. Krimper, Michael R. Lazear, and Christopher Stammen, CAD : Principles and Applications, *Englewood Cliffs, NJ : Prentice-Hall Inc.*, 1985.
- [Benoist 94] Daniel Benoist, Yves Tourbier, Sandrine Germain Tourbier, *Plans d'expériences : construction et analyse*, Technique & Documentation - Lavoisier, 1994.
- [Besnard 01] Gwénola Besnard, Laboratoire de Génie Électrique virtuel : Réalisation d'un module de visualisation, Stage de DEA, Laboratoire d'Électrotechnique de Grenoble, ENSIEG-INPG, 2001
- [Bowen R] R.Bowen Loftin, MarkEngelberg, Robin Benedetti, *Virtual Controls in Interactive Environments : A Virtual Physics Laboratory*, VETL projects, [www.vetl.uh.edu/](http://www.vetl.uh.edu/).
- [Buret 96] F. Buret, D.Muller, A. Nicolas, L. Nicolas, *A CAE Tool to understand Magnetostatics*, Proceedings of 7<sup>th</sup> International IGTE Symposium, pp. 282-285, Graz (Austria) , September 1996.
- [Chevaillier 99] Pierre Chevaillier, Fabrice Harrouet, Patrick Reignier, Jacques Tisseau, *Virtual Reality en Multi-Agent Systems for Manufacturing System Interactive Prototyping*, International Scientific Workshop of Virtual Reality, Laval (France) , June 1999.
- [Coulomb 85] Jean-Louis Coulomb, Jean-Claude Sabonnadière, CAO en Electrotechnique, *Hermes Publishing*, 1985.
- [Coutaz 87] J. Coutaz, *PAC, an Object Oriented Model for Dialog Design*, Proceedings of Interac'87, pp. 431-436, Stuttgart, 1987
- [Coutaz 88] J. Coutaz, Interface Homme – Ordinateur : Conception et Réalisation, *Thèse de doctorat*, L'université Joseph Fourir, Décembre 1988.

- [Coutaz 98] Joëlle Coutaz, Modélisation en Interaction Homme-Machine, *Cours de DEA*, CLIPS-IMAG, 1998-1999.
- [CSA 00] CSA Newsletter, *Virtual Reality and CAD – Complementary or Competing*, the winter 2000 issue, vol.XII, no.3.
- [David 71] M. David Prince, Interactive Graphics for Computer-Aided Design, *Addison Wesley Publishing Company*, 1971.
- [Dede CH] Chris Dede, Marlyn C. Salzman, R. Bowen Loftin, *ScienceSpace : Virtual Realities for Learning Complex and Abstract Scientific Concepts*, VETL projects, [www.vetl.uh.edu/](http://www.vetl.uh.edu/).
- [Deonsant 96] Jaques Deonsant, *Comprendre et mener des plans d'expériences*, AFNOR 1996.
- [Dewi 98] Alita Dewi, *Génie Electrique Virtuel*, Mémoire de DEA, Collègue doctorat de l'INPG, Laboratoire de l' Electrotechnique de Grenoble, Juin 1998.
- [Dular 98] P. Dular, C. Geuzaine, F. Henrotte, W. Legros, A general environment for the treatment of discrete problems and its application to the finite element method, *IEEE Transactions on Magnetism*, Vol.34, No.5, pp. 3395-3398, 1998.
- [Encarnacao 87] Encarnacao, Jose, Rolf Linder, an Ernst G. Schlechtendahl, ed., *Computer Aided Design: Fundamental and System Architectures*, *Springer-Verlag*, Berlin, 1987.
- [Fauchais 90] Pierre Fauchais, « Plasma Thermique : Production et applications », *Technique de l'ingénieur*, traité Génie électrique, D2 II, pp. D 2 820-5 – D 2 820-10, 1990.
- [Féchat 95] Louis Féchat, *Le contact électrique : appareillage de connexion*, pp.305 - 354, Paris : Hermes, 1995.
- [Féchat 84] Louis Féchant, "Appareillage électrique à base tension", *Techniques de l'ingénieur*, traité Génie électrique, D5 I, pp. D 4 860-1 – D 4862-14, 1984.
- [Ferber 95] Ferber J., *Les systèmes Multi-agents : vers une intelligence collective*, InterEditions, 1995.
- [Fournet 93] Gérard Fournet, « Electromagnétisme », *Technique de l'ingénieur*, traité Génie électrique, DI, pp. D 1 025-19 – D 1 025-26, 1993.

- 
- [Franklin 96] Stan Franklin and Art Graesser, *Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents*, Proceedings of the Third International Workshop on Agent Theories, Architecture, and Language, 1996.
- [Gardan 83] Yvon Gardan, Michel Lucas, *Technique Graphiques Interactives et CAO*, *Hermes Publishing*, 1983.
- [Gentil 88] Alain Gentilhomme, C.O.C.A.S.E Un système expert d'aide à la conception d'appareillages électriques, *Thèse de doctorat*, Laboratoire d'Electrotechnique de Grenoble, ENSIEG-INPG, 1988.
- [Hadipriono 96] Fabian C. Hadipriono, "Virtual Reality Application in Civil Engineering", *VRST'96 Proceedings*, pp.93 – 100, July 1996.
- [Hoole 89] Hoole, S. Ratnajeevan H., *Computer-Aided Analysis and Design of Electromagnetic Devices*, *Elsevier*, New York, 1989.
- [Isdale 93] Jerry Isdale, *What Is Virtual Reality ?*, Homebrew Introduction and Information Resources List, version 2.1, October 1993.
- [ITNews 99] IT News, *Computer and Ship*, Edition August 1999.
- [Jacobson 94] Jacobson, R, *Virtual worlds: A new type of design environment*, *Virtual Reality World*, pp. 45 -52, May/June 1994.
- [Leathrum 00] James F. Leathrum, Jr., Roland R.Mielke, and Taylor Frith, *An Integrated Spreadsheet Approach to Simulation Interfaces*, the proceedings of the 2000 SCSC, Vancouver–British Columbia, July 2000.
- [Leplus 92] François Leplus, "Bobine à noyau de fer en régime variable", *Technique de l'ingénieur*, traité Génie électrique, DI, pp. D 3 010-5 – D 3 010-10, 1992.
- [Maréchal 00] Yves Maréchal, Vers une Nouvelle Génération de Logiciels de Simulations pour l'Electromagnétisme et les Disciplines Connexes, *Habilitation à diriger les recherches*, LEG – ENSIEG, 2000.
- [Okayama 99] Eiji Okayama, Masakazu Muraki, Vlatko Cingoski, Kazufumi Kaneda and Hideo Yamashita, *Interactive Visualisation System for Education and Design in Electromagnetics*, Proceedings of Compumag, pp. 282-283, Sapporo (Japan), October 1999.

- 
- [Ouadou 94] Ouadou K, AMF : Un modèle d'architecture multi-agents multi-facettes pour Interfaces Homme – Machine et les outils associés, *Thèse de doctorat*, Ecole Centrale de Lyon, 1994.
- [Oren 78] Tuncer.I.Oren, "Concepts for Advanced Computer Assisted Modelling", *Proceeding of Symposium on Modelling end Simulation Methodology*, pp.29-55, Israel, August 1978.
- [Picard 96] Pierre Picard, "Essais en électricité", *Technique de l'ingénieur*, traité Génie électrique, D2 I, pp. D2 020-1 - D2 020-20, 1996.
- [Pierre 96] Etienne Pierre, "Essais des machines tournantes", *Technique de l'ingénieur*, traité Génie électrique, D3 III, pp. D3750-1 - D3750-24, 1996.
- [Poquet 98] Poquet Johnny, Architecture AMF des systèmes interactifs : conception d'un moteur de gestion d'interactions en Java, *mémoire de DEA*, Université Claude Bernard de Lyon, 1998
- [Preis 94] K. Preis, I. Bardi, O. Biro, R. Hoscheck, M. Mayr, U. Peterlini, K.R. Richter, I. Ticar, *Computer Animation of Electromagnetic Phenomena*, CEFC, France, July 1994, <http://www-igte.tu-graz.ac.at/index.htm>
- [Roters 41] H.C. Roters, *Electromagnetic Devices*, John Wiley and Sons inc., New York, 1941
- [Servat 98] David Servat, Edith Perrier, Jean-Pierre Treuil, and Alexis Drogoul, *Towards Virtual Experiment Laboratories : How Multi-Agent Simulations Can Cope with Multiple Scales of Analysis and Viewpoints*, *Proceedings of Virtual Worlds 98*, pp. 205-217, France, July 1998.
- [Simon 68] H.A. Simon, "The science of Artificial", *MIT Press*, 1968.
- [Sowizral 97] Henry Sowizral, KevinRushforth, Michael Deering, *The Java 3D API Specification*, Addison-Wesley, December 1997.
- [Tarpin 99] Franck Tarpin - Bernard, Bertrand David, *AMF : un modèle d'architecture multi-agents multi-facettes*, *Technique et sciences informatiques*, pp. 555-586, vol. 18, no.5, Mai1999.
- [Telemecanique 91] Telemecanique, Catalogue Avril 1991, Groupe Schneider.
- [Thollon 99] F. Thollon, L. Nicolas, *A computer Aided Education Tool for Electromagnetic Scattering Phenomena*, *Proceedings of Compumag*, pp. 8-9, Sapporo (Japan), October 1999.
-

- 
- [Treuil 97] Jean Pierre Treuil, Edith Perrier, Christophe Cambier, *Directions pour une approche multi-agents de la simulation de processus physiques spatialisés*, Proceedings of JFIADSMA'97, pp. 211-227, 1997.
- [Vacquié 96] Serge Vacquié, "Arc électrique", *Technique de l'ingénieur*, traité Génie électrique, D2 II, pp. D2 870-1 – D 2 870-26, 1996.
- [VRML 97] The VRML Consortium Inc., The Virtual Reality Modelling Language, *International Standard ISO/IEC 14772-1:1997*, Copyright 1997.
- [Youngblut 96] Christine Youngblut, Rob E. Johson, Sarah H. Nash, Ruth A. Wienclaw, and Craig A. Will, "Review of Virtual Environment Interface Technology", IDA Paper P-3186, Institute for Defense Analyses, March 1996.
- [Youngblut 98] Chistine Youngblut, *Educational Uses of Virtual Reality*, IDA Document D-2128, Institute for Defense Analyses, January 1998.



## **Annexes**





## ANNEXE A : METHODOLOGIE DE LA MODELISATION DE LABORATOIRE D'EXPERIMENTATION EN ELECTROTECHNIQUE

Les étapes		Sous étapes	
	Description (résultats)		Description (résultats)
Formaliser la définition de l'expérimentation de dispositifs électriques	Observer le fonctionnement et les comportements physiques d'un dispositif électrique en mettant en place le dispositif dans les conditions adéquates		
Formaliser le type de l'expérimentation	1. Expérimentation non normalisée (expérimentation de nouveau dispositif) 2. Expérimentation normalisée (expérimentation de dispositif connu)		
Formaliser les entités dans une expérimentation	1. Expérimentateur	Identifier les composants de chaque entité	
	2. Schéma de la manipulation d'expérimentation		a. Dispositifs électriques b. Instruments de mesure c. Source électriques
	3. Normes de l'expérimentation		a. Hypothèses b. Contraintes c. Protocole d'expérimentation
Formaliser les activités (le processus d'expérimentation)	1. Définir le problème étudié	Identifier la séquence des étapes de chaque activité	a. Déterminer le dispositif électrique étudié b. Déterminer l'objectif d'expérimentation
	2. Développer la manipulation (modèle d'expérimentation)		a. Développer le schéma de la manipulation d'expérimentation b. Créer l'état dans lequel le dispositif doit opérer
	3. Mettre en oeuvre l'expérimentation		a. Déterminer les composants du dispositif qui seront observés b. Noter les caractéristiques du fonctionnement et de la géométrie du dispositif électrique c. Faire fonctionner le dispositif électrique d. Analyser les résultats obtenus e. Comparer les résultats obtenus avec les résultats théoriques f. Evaluer les performances du dispositif électrique g. Valider les performances du dispositif
	4. Observer		a. Observer les comportements physiques, soit par des instruments de mesure, soit par les réaction physiques du dispositif
	5. Analyser		a. Evaluer les résultats obtenus b. Exprimer les résultats basés sur les théories relatives c. Tirer la conclusion

Les étapes		Sous étapes		
	Description (résultats)		Description (résultats)	
Formaliser les connaissances	1. Connaissance descriptive	Définir les connaissances	Dispositif électrique : a. Géométrie b. Matériau c. Caractéristiques de fonctionnement	Décomposer les paramètres et modéliser le fonctionnement et les comportements physiques
	2. Connaissance déductive		Normes d'expérimentation	
	3. Connaissance stratégique		Contrôle des activités lors de l'expérimentation	



Les étapes de simulation qui sont assimilées aux processus se produisant dans le dispositif électrique

## ANNEXE B : CATEGORIES DE DISPOSITIFS ELECTRIQUES

No.	Catégories des dispositifs électriques	Sous Catégories			
		<i>Forme de l'électroaimant</i>	<i>Courant de fonctionnement</i>		
1	Appareillage	en "U"	contacteur		
2			relais		
3		en "E"	contacteur		
4			relais		
5		plongeant	contacteur		
6			relais		
	Transformateur	<i>Enroulement</i>	<i>Phase</i>	<i>Couple de bobine</i>	
7		2 enroulements	monophasé		
8			trois phases	étoile-étoile	
9				étoile-triangle	
10				étoile-zigzag	
11				triangle-étoile	
12				triangle-triangle	
13				triangle-zigzag	
14		autotransformateur	monophasé		
15			trois phases	étoile-étoile	
16		3 enroulements	monophasé		
17			trois phases	étoile-étoile-étoile	
18				étoile-étoile-triangle	
19				étoile-étoile-zigzag	
20				triangle-triangle-étoile	
21				triangle-triangle-triangle	
22				triangle-triangle-zigzag	
	Machine Tournante	<i>Type d'alimentation</i>	<i>Circuit d'excitation</i>	<i>Position du circuit d'excitation</i>	<i>Fonctionnement</i>
23		Courant Continu	interne	série	Générateur
24					Moteur
25					Moteur pas à pas
26				parallèle	Générateur
27					Moteur
28					Moteur pas à pas
29			externe (indépendant)	compound	Générateur
30					Moteur
31					Moteur pas à pas
32					Générateur
33					Moteur
34					Moteur pas à pas
		Courant Alternatif	<i>Couple de rotor-stator</i>	<i>Aimant d'excitation</i>	<i>Fonctionnement</i>
35			synchrone	permanent	Générateur
36					Moteur
37					Moteur pas à pas
38			asynchrone	électroaimant	Générateur
39					Moteur
40					Moteur pas à pas
41					Générateur
42					Moteur
43					Moteur pas à pas

# **ANNEXE C : COMPORTEMENTS PHYSIQUES D'UN CONTACTEUR ELECTROMECHANIQUE**

Condition de contacteur	Alimentation de la bobine	Comportements physiques	
		Position des pôles de contacteur	Performance observée
transitoire mécanique	courant alternatif courant continu	position ouverte (repos)	
		↓	temps d'établissement du courant (alimentation de la tension)
		↓	courant d'engrènement (lorsque le contacteur commence à fermer) force magnétique
		point de début de la fermeture	vitesse de déplacement
		↓	variation de l'inductance
		↓	la chute de courant
		↓	temps de fermeture
		↓	courant de maintien
		↓	pertes Joule
		↓	pertes hystérésis (aimant non linéaire)
		position fermée	choc mécanique des pôles (si alimentation en courant alternatif)
		↓	courants de Foucault
		↓	effet des spires de Frager
		↓	diminuer la tension d'alimentation
		point de début de l'ouverture	courant de déclenchement (lorsque le contacteur commence à ouvrir)
		↓	force magnétique
		↓	vitesse de déplacement de l'armature
		↓	arc électrique
		↓	temps d'ouverture
		↓	force résiduelle
		↓	induction rémanente
		position ouverte (repos)	

Condition de fonctionnement	Alimentation de la bobine	Comportements physiques	
		Position des pôles de contacteur	Performance observée
permanent mécanique		position ouverte position repos	force (d'appel) tension (d'appel) courant (d'appel) courant de Foucault pertes Joule pertes hystérésis (aimant non linéaire) effet des spires de Frager
	courant alternatif courant continu	vers fermé et position fermée	force (de maintien) tension (de maintien) courant (de maintien) courant de Foucault pertes de Joule pertes hystérésis ( non linéaire) effet des spires de Frager

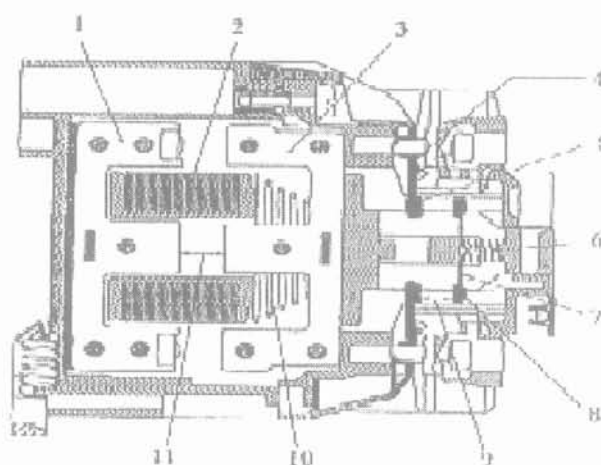
## ANNEXE D : MODELES MATHEMATIQUES DES COMPORTEMENTS PHYSIQUES D'UN CONTACTEUR ELECTROMECHANIQUE

### D.1. INTRODUCTION

Un contacteur est un interrupteur commandé par un électroaimant. Il est utilisé pour isoler le dispositif de commande du dispositif de puissance. Un contacteur est constitué par un certain nombre d'éléments qui sont :

1. les pôles qui assurent l'établissement et l'interruption des courants principaux,
2. l'électroaimant qui est l'organe moteur du contacteur ou un système générateur de champ magnétique créé par un bobinage fixe, associé à un circuit magnétique,
3. le support,
4. les contacts auxiliaires et les contacts temporisés.

Concernant la forme du circuit magnétique d'un électroaimant de contacteur, elle peut être : en forme de U, avec l'armature mobile plongeante ou l'armature mobile plate, en forme de E, avec aussi l'armature mobile plongeante ou l'armature mobile plate, ou en forme de noyau plongeant. La figure D-1 présente un contacteur possédant un électroaimant en forme de E avec une armature mobile plongeante.



1. Armature fixe de l'électroaimant
2. Bobine de l'électroaimant
3. Armature mobile de l'électroaimant
4. Support de contact fixe
5. Rivet de contact fixe
6. Ressort de pôle
7. Support des contacts mobiles (CuSn6P)
8. Rivet de contact mobile
9. Course d'ouverture des contacts
10. Ressort de rappel de l'électroaimant
11. Course de l'électroaimant

Figure D - 1 Contacteur électromécanique

Le principe de fonctionnement d'un contacteur est simple : une excitation, souvent faible, permet de déplacer des parties conductrices (des contacts) qui, lorsqu'elles entreront en contact, feront transiter une puissance beaucoup plus importante [Gentil 88]. Les cas limites du fonctionnement de l'électroaimant d'un contacteur sont, soit à courant constant dans le bobinage, soit à flux constant dans le circuit magnétique.

Les comportements physiques intervenant pendant le fonctionnement d'un contacteur électromécanique seront présentés par les équations mathématiques ci-dessous.

## D.2. FORCE

La force qui intervient durant le fonctionnement d'un contacteur peut être divisée en force magnétique générée par l'électroaimant et force de contacts générée par la striction des lignes de courant. Leurs modèles mathématiques sont les suivants :

### A. Force magnétique ou force d'attraction

#### 1. Alimentation en courant continue

La force d'attraction sur l'entrefer est

$$F = \frac{B^2 S}{2\mu_0} \quad (F.1)$$

avec l'induction magnétique :

$$B = \frac{\mu_0 Ni}{l} \quad (F.2)$$

#### 2. Alimentation en courant alternatif

Lorsque la bobine est alimentée par un courant alternatif, l'induction magnétique est générée à la même fréquence que la source :

$$B = B_m \sin \omega t \quad (F.3)$$

En se basant sur l'équation (F.1), la force d'attraction devient :



$$F = \frac{B_m^2 \sin^2 \omega t}{2\mu_o} S \quad (F.4)$$

$$F = S \left( \frac{B_m^2}{4\mu_o} - \frac{B_m^2 \cos 2\omega t}{4\mu_o} \right) \quad (F.5)$$

L'équation (F.5) montre que cette force se compose de deux éléments : la force utile pour déplacer l'armature et la force de pulsation qui provoque l'annulation de la force utile deux fois par période. Les expressions de ces forces sont:

$$F_u = S \frac{B_m^2}{4\mu_o} \quad (F.5.a)$$

$$F_p = -S \frac{B_m^2 \cos 2\omega t}{4\mu_o} \quad (F.5.b)$$

où :

B : est l'induction magnétique (Tesla),

F : est la force d'attraction (N),

$F_u$  : est la force utile (N),

$F_p$  : est la force de pulsation (N),

N : est le nombre de tours de la bobine,

i : est le courant de la bobine (A),

S : est la surface traversée par le flux magnétique (m<sup>2</sup>),

$\mu_o$  : est la perméabilité de l'entrefer ,

l : est la dimension de l'entrefer (m),

$B_m$  : est l'induction maximum (Tesla),

$\omega$  : est la fréquence des courants électriques (Hertz),

t : est le temps (seconde).

## B. Force des contacts

La force des contacts est essentiellement la force de répulsion due aux striction. En effet, le courant conduit par les contacts ne passe pas par toute leur surface. Il existe une striction des lignes de courant qui provoque une répulsion. Cette striction limite la tenue aux surcharges des contacts électriques [Féchat 84]. La répartition des lignes de courant est semblable dans la striction et dans les sections à l'infini, à celle montrée sur la figure 2.

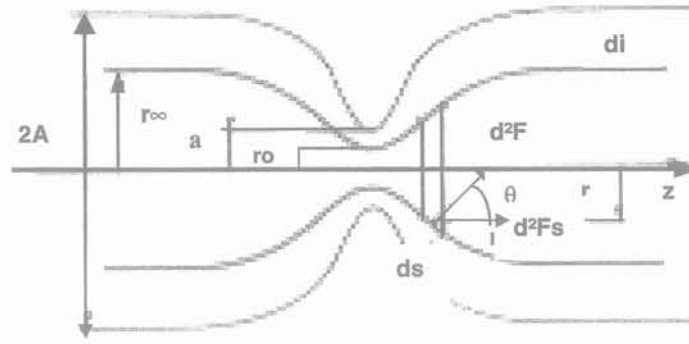


Figure D - 2 Répulsion due à la striction

Si l'on considère un conducteur, l'induction induite par le courant peut être prise égale à:

$$B = \frac{\mu_o \cdot i}{2\pi r} \quad (\text{F.6})$$

La force totale du courant est:

$$d^2F = \frac{\mu_o \cdot i}{2\pi r} di \cdot ds \cdot \cos \theta = \frac{\mu_o \cdot i}{2\pi r} di \cdot dr \quad (\text{F.7})$$

et la force de répulsion, ou de striction, est :

$$dF_s = \frac{\mu_o \cdot i \cdot di}{2\pi} \ln[r]_{r_o}^{r_{\infty}} = \frac{\mu_o \cdot i \cdot di}{2\pi} \ln\left[\frac{r_{\infty}}{r_o}\right] \quad (\text{F.8})$$

Pour une distribution uniforme dans la striction, on peut déduire:

$$F_s = \frac{\mu_o \cdot i^2}{4\pi} \ln\left[\frac{2A}{2a}\right] \quad (\text{F.9})$$

où :

$a$  : est le rayon de striction,

$r$  : est le rayon du tube de courant variant de  $r_o$  à la striction à  $r_{\infty}$ ,

$2A$  : est le diamètre moyen de la pastille de contact.,

$F_s$  : est la force de répulsion,

$i$  : est le courant conduit par les contacts.

A part la force de répulsion, il existe aussi les forces en présence de masses ferromagnétiques. Les pièces ferromagnétiques sont fréquemment utilisées pour augmenter les forces sur les courants. Ceci permet d'éviter la répulsion des contacts en cas de surcharge ou, à l'inverse, de provoquer une ouverture rapide en cas de court-circuit.

### D.3. ENERGIE

La force d'attraction est déduite de l'énergie stockée dans l'entrefer par dérivation :

$$F = \frac{dW}{dl} \quad (\text{E.1})$$

En utilisant l'équation (F.1), l'énergie stockée dans l'entrefer s'écrit :

$$W = B^2 S \frac{l}{2\mu_0} \quad (\text{E.2})$$

### D.4. SPIRES DE FRAGER

#### A. Principe [Roters 41]

Lorsque la bobine est alimentée en courant alternatif, le flux induit est lui aussi alternatif. Ceci provoque une pulsation dans la force (§ D.2) qui n'est pas souhaitable en position fermée (l'armature mobile étant en contact avec l'armature fixe). Une bague de déphasage, appelée spire de Frager, est donc montée sur une branche du circuit magnétique pour atténuer la force de pulsation. Elle entoure environ les deux tiers de la surface de cette branche (figure D-3). Cette spire, en court circuit sur elle-même, a pour effet de décaler en arrière le flux qui la traverse. De cette façon, ce flux  $\phi_2$  est déphasé par rapport à celui  $\phi_1$  qui traverse la surface non entourée par la spire (figure D-4). La somme de ces deux flux n'est jamais égale à zéro. Il en résulte donc une force d'attraction qui ne s'annule jamais.

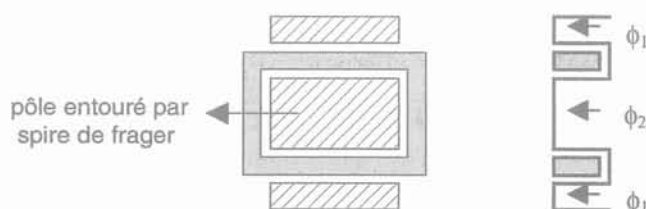


Figure D - 3 Section de circuit magnétique concernée par la spire de Frager

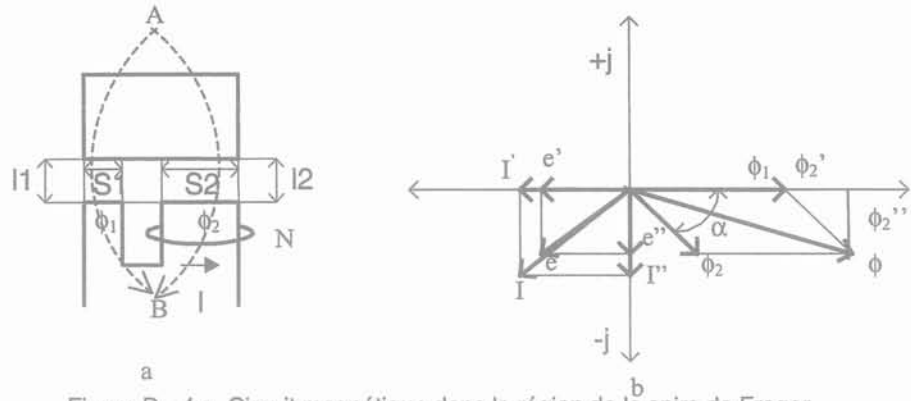


Figure D - 4 a. Circuit magnétique dans la région de la spire de Frager,  
b. Diagramme vecteur des flux, des tensions et des courants induits dans la spire de Frager

Les réluctances dans l'entrefer sur la surface entourée par la spire de Frager et sur celle non entourée, sont définies par l'équation suivante :

$$\mathcal{R} = \frac{l}{\mu_0 S} \quad (\text{S.1})$$

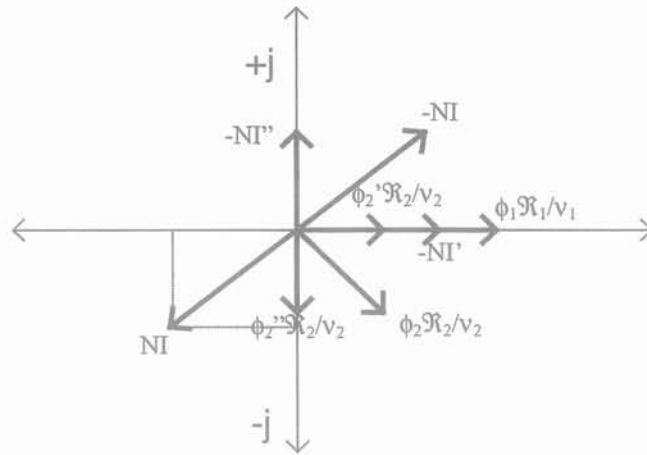


Figure D - 5 Diagramme vecteur des forces magnétomotrices de la surface de pôle aimant entourée par les spires de Frager et de celle non entourée.

La force magnétomotrice entre les points A et B (figure 4.a) vérifie l'équation vectorielle (figure 5) suivante:

$$\phi_1 \frac{\mathcal{R}_1}{V_1} = \phi_2 \frac{\mathcal{R}_2}{V_2} - NI \quad (\text{S.2})$$

Les composantes réelles de la force magnétomotrice vérifient :

$$\phi_1 \frac{\mathcal{R}_1}{V_1} = \phi_2 \frac{\mathcal{R}_2}{V_2} NI \quad (\text{S.3})$$

Avec le courant induit dans la spire de Frager :

$$I = \frac{e^i}{r} = \frac{\omega N^2 \phi_2''}{r} \quad (\text{S.4})$$

et le taux de réactance par rapport à la résistance :

$$k = \frac{V_2 \omega N^2}{\mathcal{R}_2 r} \quad (\text{S.5})$$

l'équation (S.3) devient :

$$\phi_1 \frac{\mathcal{R}_1 V_2}{V_1 \mathcal{R}_2} = \phi_2' + k \phi_2'' \quad (\text{S.6})$$

Les composants imaginaires de la force magnétomotrice (équation (S.2)) :

$$-NI'' = -(-j\phi_2'' \frac{R_2}{V_2}) \quad (\text{S.7})$$

avec le courant imaginaire :

$$I'' = \frac{e''}{r} = \frac{-j\omega N \phi_2'}{r} \quad (\text{S.8})$$

En substituant l'équation (S.5) et l'équation (S.8) sur l'équation (S.7), on obtient :

$$k\phi_2' = \phi_2'' \quad (\text{S.9})$$

En résumé, on peut alors déduire le flux déphasé  $\phi_2$  et l'angle  $\alpha$  entre le flux  $\phi_1$  et le flux  $\phi_2$  :

$$\phi_2' = \phi_1 \frac{1}{1+k^2} \frac{\mathcal{R}_1 V_2}{\mathcal{R}_2 V_1} \quad (\text{S.10})$$

$$\phi_2'' = \phi_1 \frac{k}{1+k^2} \frac{\mathcal{R}_1 V_2}{\mathcal{R}_2 V_1} \quad (\text{S.11})$$

$$\phi_2 = \sqrt{(\phi_2')^2 + (\phi_2'')^2} \quad (\text{S.12})$$

$$\tan \alpha = \frac{\phi_2''}{\phi_2} = k \quad (\text{S.13})$$

où :

$\phi_1$  : est le flux de pôle non entouré par la spire de Frager,

$\phi_2$  : est le flux de pôle entouré par la spire de Frager,

$\alpha$  : est l'angle entre flux  $\phi_1$  et flux  $\phi_2$ ,

$\mathfrak{R}$  : est la réluctance,

$v$  : est le coefficient de fuites,

$r$  : est la résistance de la spire de Frager,

$I$  : est le courant induit dans la spire de Frager,

$N$  : est le nombre de tours de la spire de Frager,

$e$  : est la tension induite par le flux  $\phi_2$ .

### B. L'effet des spires de Frager sur la force d'attraction

La spire de Frager sera idéalement conçue lorsque l'angle  $\alpha$  sera égal à  $90^\circ$ . Dans ce cas la force alternative deviendra égale à zéro.

Les forces sur la surface entourée par la spire et sur celle non entourée par la spire sont :

$$F_1 = \frac{(\phi_{1m})^2 \sin^2 \omega t}{v_1^2 S_1 \cdot 2 \cdot \mu_0} \quad (\text{S.14})$$

$$F_2 = \frac{(\phi_{2m})^2 \sin^2(\omega t - \alpha)}{v_2^2 S_2 \cdot 2 \cdot \mu_0} \quad (\text{S.15})$$

où le flux  $\phi_1$  maximum et le flux  $\phi_2$  maximum :

$$\phi_{1m} = \phi_1 \cdot \sqrt{2} \quad (\text{S.16})$$

$$\phi_{2m} = \phi_2 \cdot \sqrt{2} \quad (\text{S.17})$$

La force d'attraction est donc la somme des équations (S.14) et (S.15) :

$$F = \frac{(\phi_{1m})^2 \sin^2 \omega t}{v_1^2 S_1 \cdot 2 \cdot \mu_0} + \frac{(\phi_{2m})^2 \sin^2(\omega t - \alpha)}{v_2^2 S_2 \cdot 2 \cdot \mu_0} \quad (\text{S.18})$$

$$F = \frac{\phi^2}{2\mu_0 v_1^2 S_1} + \frac{\phi^2}{2\mu_0 v_2^2 S_2} \left( \frac{\phi^2 \cos 2\omega t}{2\mu_0 S_1 v_1^2} + \frac{\phi^2 \cos(2\omega t - 2\alpha)}{2\mu_0 v_2^2 S_2} \right) \quad (\text{S.19})$$

L'équation (S.19) montre que la force d'attraction se constitue d'une composante continue et d'une composante alternative. En plus, elle sera continue lorsque  $\alpha$  sera égal à  $90^\circ$ .

## D.5. PERTES MAGNETIQUES

### A. Hystérésis [Leplus 92]

Les pertes par hystérésis sont séparées en deux cas suivant la valeur de l'Induction magnétique maximum  $B_{\max}$  :

Pour  $B_{\max} < 0.2$  T, la représentation de Rayleigh conduit à la formulation :

$$P_H = k_R V_{CM} f H_{\max}^3 \quad (M.1)$$

Et pour  $0.2T < B_{\max} < 1.5$  T, on calcule les pertes par hystérésis par la formule de Steinmetz :

$$P_H = k_H V_{CM} f B_{\max}^\eta \quad (M.2)$$

avec

$$k_H = \cos \varphi \frac{\pi H_{\max}}{(B_{\max})^{\eta-1}} \quad (M.3)$$

où :

$P_H$  : sont les pertes par hystérésis,

$k_R$  : est le coefficient caractérisant le matériau,

$\eta$  : est le coefficient de pertes par hystérésis ( $1.3 \leq \eta \leq 2.2$ ),

$\psi$  : est le déphasage arrière du courant,

$f$  : est la fréquence,

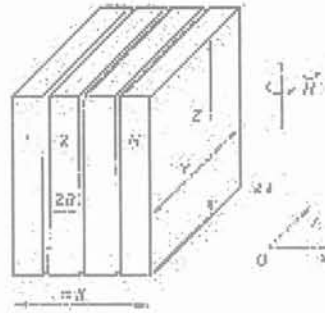
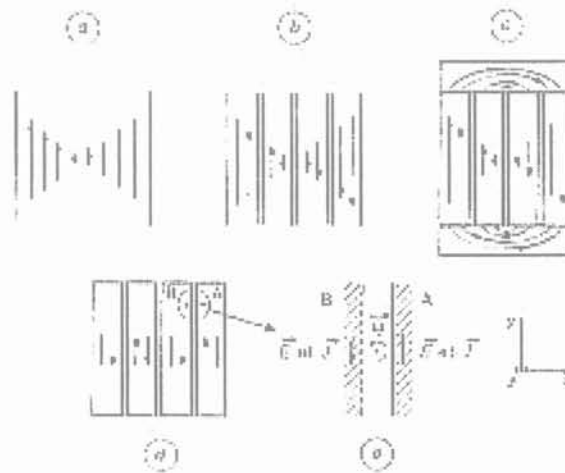
$V_{CM}$  : est le volume de circuit magnétique.

### B. Perte par courants de Foucault [Fournet 93]

Nous considérons que le tôle est un ensemble de plaques ou un paquet de tôles (figure D-6). Ces plaques sont quasi jointives, isolées les unes des autres par une couche isolante dont l'épaisseur est négligeable devant  $2a$  (l'épaisseur d'une tôle).

On suppose, pour idéaliser le problème :

- que le champ magnétique n'a qu'une composante  $H_z$  ( $H_z(x,t)$ ),
- que les effets de bord sont négligeables lorsque  $Y \gg 2a$  et  $Z \gg 2a$ ,
- que le champ électrique n'aura donc qu'une composante  $E_y$  ( $E_y(x,t)$ ).

Figure D - 6 Un paquet de tôles ( $X \approx 2Na$ )

Les flèches donnent symboliquement la direction et le module de la densité des courants de Foucault, soit dans une tôle de grande épaisseur (a), soit dans un groupe de 4 tôles [(b), (c), (d)] avec divers environnements. La partie (e) de la figure concerne un détail de la partie (d).

Figure D - 7 Courants de Foucault

En rapportant les pertes par courant de Foucault à **la moyenne spatiale de l'induction**, la représentation complexe de l'induction est définie par :

$$\langle B \rangle_x = \frac{kU}{j\omega xy} \quad (\text{M.4})$$

Lorsque la fréquence de courant de la bobine ( $\omega$ ) est très inférieure à la fréquence de Foucault ( $\omega_F$ ), la puissance dissipée par courant de Foucault correspondant aux formes asymptotiques de la fonction  $\alpha$  est :



$$\langle P \rangle_x = \frac{xyz}{8a^2 \mu^2} |\langle B \rangle_x|^2 \frac{\alpha^4}{3} \quad (\text{M.5})$$

Au contraire si, si  $\omega$  est très supérieure  $\omega_F$  on a :

$$\langle P \rangle_x = \frac{xyz}{8a^2 \mu^2} |\langle B \rangle_x|^2 \alpha^3 \quad (\text{M.5})$$

La fréquence de Foucault ( $\omega_F$ ), utilisée ci-dessus, est définie par:

$$\omega_F = \frac{9}{2a^2 \mu \gamma} \quad (\text{M.6})$$

$$\alpha = a \sqrt{2 \omega \mu \gamma} \quad (\text{M.7})$$

où :

$\mu$  : est la perméabilité de tôles,

$\gamma$  : est la conductivité de tôles,

$2a$  : est la largeur de tôle,

$k$  : est la fraction du flux d'induction par rapport le flux total,

$U$  : est la tension par spire de la bobine.

### C. Courants de Foucault [Leplus 92]

Les tôles parcourues par les courants de Foucault sont assimilées à des spires en court-circuit couplées au bobinage principale (figure D-8)

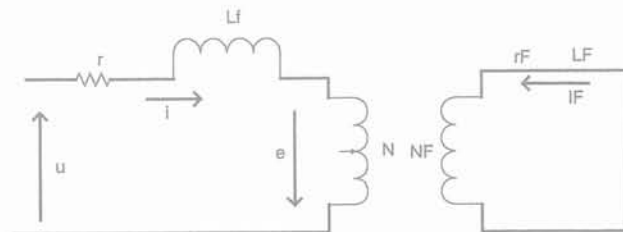


Figure D - 8 Schéma équivalent des tôles parcourues par les courants de Foucault

Pour les spires en court circuit, au regard du schéma équivalent ci-dessus, on peut écrire la formule :

$$0 = i_F r_F + L_F \frac{di_F}{dt} + N_F \frac{d\phi}{dt} \quad (\text{M.8})$$

et pour l'enroulement principal (la bobine) :

$$u = ir + L_f \frac{di}{dt} + N \frac{d\phi}{dt} \quad (\text{M.9})$$

En considérant le théorème d'Ampère, on obtient l'équation de couplage entre la bobine et la spire :

$$N_F i_F + N i = H l_{CM} \quad (\text{M.10})$$

Ainsi, les courants de Foucault s'écrivent:

$$i_F = \frac{H l_{CM} - N i}{N_F} \quad (\text{M.11})$$

où :

$r_F$  : est la résistance l'ensemble des tôles au trajet des courants de Foucault,

$L_F$  : est l'inductance de fuites par rapport à l'enroulement principale (bobine),

$N_F$  : est le nombre de spires en court circuit (nombre de trajets des courants de Foucault),

$N$  : est le nombre de spires de la bobine,

$i_F$  : est la valeur instantanée des courants de Foucault,

$i$  : est la valeur instantanée du courant de la bobine.

#### D. Trajets des courants de Foucault:

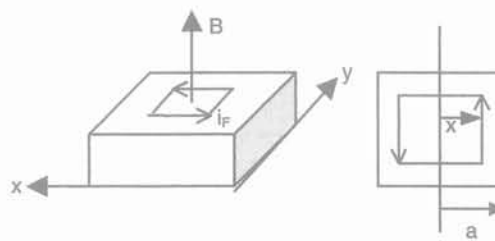


Figure D - 9 Profil des trajets des courants de Foucault

Les courants de Foucault à la distance  $x$  par rapport à l'inductance se formule :

$$i_F = \frac{x}{\rho} \frac{dB_{\max}}{dt} = k_F f B_{\max} x \quad (\text{M.12})$$

où:

$\rho$  : est la résistivité du circuit magnétique.

Nous utilisons l'équation (M.12) pour dessiner les courants de Foucault parcourant la tôle.

## D.6. COURANT DE LA BOBINE

Lorsque la bobine est alimentée par la source de courant continu, le courant conduit par la bobine s'établit au bout d'un certain temps qui dépend du matériau de la bobine. En plus, durant le déplacement de l'armature mobile, la variation de l'inductance s'oppose au passage du courant absorbé par la bobine. Les deux phénomènes sont définis par les équations suivantes :

### A. Temps d'établissement (à la fermeture)

$$i = \frac{E}{R} \left( 1 - e^{-\frac{R}{L}t} \right) \quad (C.1)$$

$$t = -\frac{L}{R} \ln \left( 1 - \frac{R \cdot i}{E} \right) \quad (C.2)$$

### B. La variation de l'inductance

$$\frac{dL}{dl} = -\frac{\mu_o N^2 S}{l^2} \quad (C.3)$$

$$L = \frac{N^2 \mu_o S}{l_p} \quad (C.4)$$

$$L = \frac{N^2 \mu_o S}{(l - v.t)} \quad (C.5)$$

où :

L : est l'inductance,

v : est la vitesse de l'armature,

t : est le temps de déplacement de l'armature,

l : est l'entrefer initial,

$l_p$  : est l'entrefer en cours,

S : est la surface de l'entrefer,

$\mu_o$  : est la perméabilité de l'entrefer.

## D.7. TEMPERATURE DE LA BOBINE [Roters 41]

Au regard de la température, les spires de la bobine ont deux propriétés relatives : la capacité calorifique et la capacité de dissipation de la chaleur. En effet, les spires peuvent stocker l'énergie

thermique ce qui augmente leur température. En plus, elles sont capables de conduire de l'énergie thermique et de la dissiper par leur surface en fonction de la différence de température entre cette surface et leur environnement.

Les deux capacités apparaissent dans la formule suivante :

$$P = C \frac{d\theta}{dt} + K\theta \quad (\text{T.1})$$

où :

P : est la puissance apportée à la bobine (Watt),

C : la capacité thermique (J/kg.°C),

K : est la capacité de dissipation de la chaleur (Watt/°C),

$\theta$  : est la différence entre la température moyenne de la bobine et celle de l'environnement (°C),

t : est la période d'alimentation de la bobine (seconde).

En simplifiant l'équation (T.1), on obtient :

$$\frac{Kd\theta}{P - K\theta} = \frac{Kdt}{C} \quad (\text{T.2})$$

L'intégration de l'équation (T.2) donne :

$$-\ln(P - K\theta) = \frac{K}{C}t + C_1 \quad (\text{T.3})$$

En considérant que lorsque  $t = 0$ , nous avons  $\theta = 0$ ,  $C_1$  est donc :

$$C_1 = -\ln P \quad (\text{T.4})$$

et l'équation (T.3) devient :

$$-\ln(P - K\theta) + \ln P = \frac{K}{C}t \quad (\text{T.5})$$

$$\ln \frac{P - K\theta}{P} = -\frac{K}{C}t \quad (\text{T.6})$$

$$1 - \frac{K\theta}{P} = e^{-\frac{K}{C}t} \quad (\text{T.7})$$

En se basant sur l'équation (T.7), la différence de température est :

$$\theta = \frac{P}{K} \left( 1 - e^{-\frac{K}{C} t} \right) \quad (T.8)$$

## D.8. DEPLACEMENT DE L'ARMATURE

La capacité de l'armature mobile pour pouvoir se déplacer, à la fermeture et à l'ouverture, et le temps de déplacement sont déterminés en utilisant l'équation de l'effort magnétique suivante [Roters 41].

$$M \frac{d^2 x}{dt^2} + k_1 \frac{dx}{dt} + k_2 x = F - F_o \quad (D.1)$$

où :

$F$  : est l'effort magnétique capable de déplacer l'armature mobile,

$F_o$  : est l'effort magnétique initial (l'armature mobile dans la position initiale,  $x = 0$ ),

$M$  : est la masse de la partie mobile,

$k_1$  : est la constante de friction,

$k_2$  : est la constante du ressort,

$x$  : est la distance de déplacement,

$\frac{d^2 x}{dt^2}$  : est l'accélération de l'armature mobile,

$\frac{dx}{dt}$  : est la vitesse de l'armature mobile.

### A. Temps de déplacement

Lorsque le temps de déplacement est petit,  $k_1 \frac{dx}{dt}$  et  $k_2 x$  sont négligeables, l'équation (D.1) devient

donc :

$$M \frac{d^2 x}{dt^2} = F - F_o \quad (D.2)$$

où l'effort magnétique  $F$  en système international est :

$$F = 2,01122 (H_w l_w)^2 \frac{dP_w}{dx} \quad (D.3)$$

$H_w$  : est le champ magnétique dans l'entrefer,

$l_w$  : est l'entrefer,

$P_w$  : est la perméance de l'entrefer.

De plus, les théorèmes d'Ampère et d'Ohm donnent les équations :

$$NI = H_w l_w + H_m l_m \quad (D.4)$$

$$E = \frac{Nd\phi}{dt} + RI \quad (D.5)$$

$H_m$  : est le champ magnétique dans le circuit magnétique (en négligeant les flux de fuite,  $H_m = H_w$ ),

$l_m$  : est la ligne moyenne du circuit magnétique,

$\phi$  : est le flux total,

$I$  : est le courant de la bobine,

$R$  : est la résistance de la bobine,

$E$  : est la tension d'alimentation,

$N$  : est le nombre de spires.

En substituant l'équation (D.3) à l'équation (D.1) et en exprimant l'effort initial  $F_o$  comme l'équation de l'effort magnétique, on obtient :

$$M \frac{d^2x}{dt^2} = 2,01122 \left[ (H_w l_w)^2 - (H_w l_w)_o^2 \right] \left( \frac{dP_w}{dx} \right)_o \quad (D.6)$$

où  $\frac{dP_w}{dx}$  est constant, égal à la valeur prise lorsque  $x = 0$ .

La force magnétomotrice de l'équation (D.6) peut être simplifiée :

$$\left[ (H_w l_w)^2 - (H_w l_w)_o^2 \right] = [H_w l_w - (H_w l_w)_o] [H_w l_w + (H_w l_w)_o] \approx [H_w l_w - (H_w l_w)_o] 2(H_w l_w)_o \quad (D.7)$$

en considérant que  $[H_w l_w - (H_w l_w)_o]$  est petit. L'équation (D.6) devient donc :

$$M \frac{d^2x}{dt^2} = 4,02244 (H_w l_w)_o [H_w l_w - (H_w l_w)_o] \left( \frac{dP_w}{dx} \right)_o \quad (D.8)$$

Ensuite, en considérant le changement de flux par rapport au temps de déplacement, la différence de la force magnétomotrice est :

$$H_w l_w - (H_w l_w)_o = \frac{\phi - \phi_o}{(P_w)_o} = \frac{1}{(P_w)_o} \frac{d\phi}{dt} t \quad (D.9)$$

Des équations (D.8) et (D.9), on peut déduire l'accélération :

$$\frac{d^2x}{dt^2} = 4,02244 \left[ \frac{(H_w l_w)_o}{M(P_w)_o} \left( \frac{dP_w}{dx} \right)_o \left( \frac{d\phi}{dt} \right)_o \right] \cdot t \quad (D.10)$$

la vitesse :

$$\frac{dx}{dt} = 2,01122 \left[ \frac{(H_w l_w)_o}{M(P_w)_o} \left( \frac{dP_w}{dx} \right)_o \left( \frac{d\phi}{dt} \right)_o \right] \cdot t^2 \quad (D.11)$$

la distance de déplacement :

$$x = \frac{4,02244}{3!} \left[ \frac{(H_w l_w)_o}{M(P_w)_o} \left( \frac{dP_w}{dx} \right)_o \left( \frac{d\phi}{dt} \right)_o \right] \cdot t^3 \quad (D.12)$$

$$x = \frac{4,02244}{3!} \left[ \frac{(H_w l_w)_o}{M(P_w)_o} \left( \frac{dP_w}{dx} \right)_o \cdot \left( \frac{E - IR}{N} \right)_o \right] t^3 \quad (D.13)$$

et le temps de déplacement :

$$t = \sqrt[3]{\frac{x}{\frac{4,02244}{3!} \left[ \frac{(H_w l_w)_o}{M(P_w)_o} \left( \frac{dP_w}{dx} \right)_o \cdot \left( \frac{E - I_o R}{N} \right)_o \right]}} \quad (D.14)$$

où:

t : est le temps de déplacement,

<sub>o</sub> : la condition initiale au point où l'armature mobile va se déplacer.

Toutefois, l'intégration directe de l'équation (D.2), donne une autre équation plus simple du temps de déplacement :

$$t = \sqrt{\frac{2 \cdot M \cdot x}{F - F_o}} \quad (D.15)$$

### B. Hypothèse de déplacement [Gentil 88]

La performance finale du contacteur électromécanique se mesure par son comportement mécanique qui correspond aux forces résistantes du ressort de rappel et du ressort de pôle. La figure D-10 montre les efforts résistants en fonction du déplacement de l'armature mobile et des contacts.

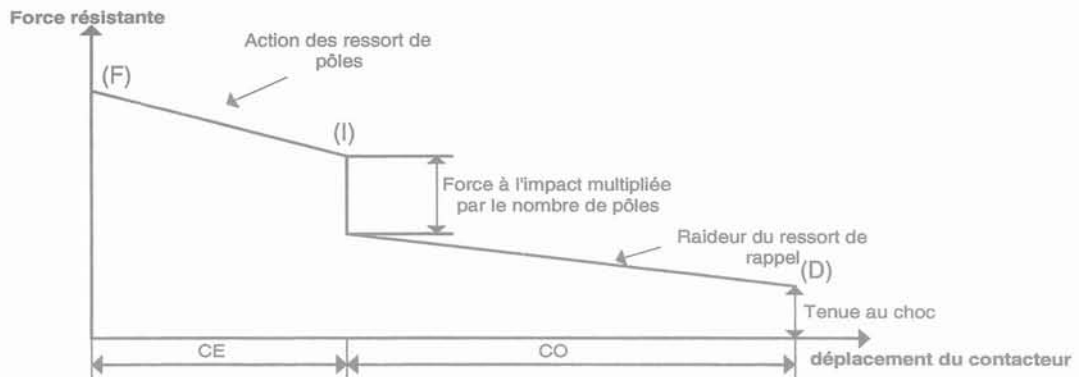


Figure D - 10 Diagramme des forces résistantes

Le diagramme comporte trois zones. A la fermeture du contacteur, en premier lieu dans la zone du démarrage (du point D au point I), l'armature mobile commence à se fermer. Cette étape fait intervenir les paramètres suivants :

- la raideur du ressort de rappel ( $r_r$ ) qui est chargé d'ouvrir le contacteur lorsque la bobine n'est plus alimentée,
- la tenue au choc du contacteur, c'est-à-dire la compression initiale du ressort de rappel qui permettra d'empêcher le déplacement de la partie mobile lors d'un choc,
- la course d'ouverture du contacteur (CO), c'est-à-dire le déplacement de l'armature mobile juste avant que les contacts mobiles touchent ceux fixes. La valeur maximum de CO est l'entrefer maximum du contacteur.

La deuxième zone (le point I) est le moment précis où les contacts mobiles et les contacts fixes se touchent. La force à l'impact correspondant à la compression initiale des ressorts de pôles augmente l'effort résistant. Cette augmentation dépend du nombre de pôles du contacteur. Enfin, dans la troisième zone (du point I au point F), les contacts mobiles se déplacent jusqu'à la fermeture complète. Les deux paramètres qui interviennent dans cette zone sont :

- la course d'écrasement (CE) des pôles qui correspond à l'épaisseur des pastilles (qui diminue au fur et à mesure de l'usure des contacts),
- la raideur des ressorts de pôles ( $r_p$ ).

### 1. A la fermeture

Le contacteur doit se fermer lorsque la tension d'alimentation atteint 0.8 – 1.1 (courant alternatif) ou 0.75 – 1.1 (courant continu) de la tension nominale. A la fermeture, un contacteur a donc besoin d'une force suffisante pour vaincre l'effort résistance ( $F_o$ ) établi par le ressort de rappel tout au long de la course de fermeture et par les ressorts de pôles tout au long de la course d'écrasement :



$$F_o = r_r \cdot (CO - 0,1 \cdot CO) + r_p \cdot CE \cdot n \quad (D.16)$$

où  $0,1CO$  est l'entrefer anti-rémanent et  $n$  est le nombre de pôles.

L'armature mobile de contacteur se déplacera quand l'effort magnétique sera plus grand que l'effort résistant;  $F > F_o$ .

## 2. A l'ouverture

Le contacteur doit retomber quand la tension d'alimentation devient inférieure ou égale à 0.1 (courant continu) ou 0.2 (courant alternatif) fois la tension nominale. Dans ce cas, l'armature mobile se déplacera à la position d'ouverture lorsque l'effort magnétique sera plus petit que l'effort résistant;  $F < F_o$ .

$$F_o = r_r \cdot (CO - 0,1 \cdot CO) \quad (D.17)$$

Concernant la course d'écrasement  $CE$ , elle diminue au fur et à mesure de l'usure des contacts. Nous considérons qu'elle est très petite, et même quasi nulle, pendant l'ouverture.

## D. 9. ARC ELECTRIQUE

Les arcs électriques ou les décharges dans l'appareillage peuvent être accidentelles ou résulter du fonctionnement normal. Dans les cas accidentels, l'origine des arcs électriques (des arcs de défaut) peut être :

- une surtension provoquée par la foudre ou par des manœuvres d'appareillage,
- la dégradation des surfaces isolantes, par cheminement ou par dépôt de poussières conductrices,
- la fissuration des isolants,
- la rupture mécanique des conducteurs ou la présence de pièces métalliques étrangères.

En plus, pour le fonctionnement normal, les arcs électriques (les arcs de coupure) se forment instantanément à l'intérieur des appareils, aux points où les contacts se séparent. Lors de l'ouverture la surface réelle des contacts se réduit. Ceci entraîne localement la fusion, et par conséquent la vaporisation du métal. Les arcs électriques peuvent alors apparaître sans surtension.

Concernant les arcs de coupure, les équations suivantes sont utilisées pour les visualiser lors du fonctionnement de contacteur.

### A. Surface réelle de contacts [Féchat 84]

La surface réelle des contacts est déterminée en calculant leur rayon de striction. Ce calcul dépend de la forme des contacts.

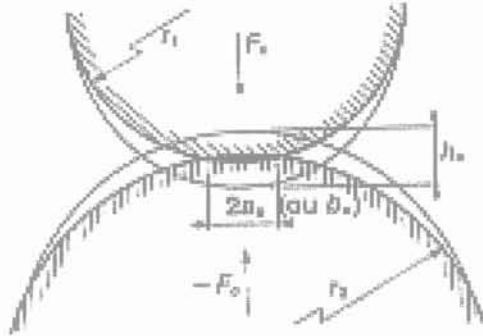


Figure D - 11 Définition de la surface réelle de contacts

#### 1. Sphères ou cylindres perpendiculaires

Le rayon de striction est en cas de déformation élastique:

$$a_e = \sqrt[3]{\frac{3}{4} \frac{F_c}{E_o \gamma_o}} \quad (\text{A.1})$$

et celui en cas de déformation plastique :

$$a_p = \sqrt{\frac{F_c}{\pi H}} \quad (\text{A.2})$$

La surface réelle de contact est donc :

$$A = \pi a_e^2 \quad (\text{A.3.1})$$

ou

$$A = \pi a_p^2 \quad (\text{A.3.2})$$

## 2. Cylindres parallèles

La largeur de striction de contact est :

$$b_e = \sqrt{\frac{4}{\pi} \frac{F_c}{E_o \gamma_o}} \quad (\text{A.4})$$

La surface réelle est alors :

$$A = b_e^2 \quad (\text{A.5})$$

Le module d'élasticité et le rayon apparent des contacts sont :

$$\frac{1}{E_o} = \frac{1-\nu_1}{E_1} + \frac{1-\nu_2}{E_2} \quad (\text{A.6})$$

$$\gamma_o = \frac{1}{r_1} + \frac{1}{r_2} \quad (\text{A.7})$$

où :

$E_1, E_2$  : sont les modules d'élasticités des contacts 1 et 2,

$F_c$  : la force de contact,

$\nu_1, \nu_2$  : les coefficients de Poisson des contacts 1 et 2 (  $0.24 < \nu < 0.44$  ),

$r_1, r_2$  : les rayons apparents des contacts 1 et 2.

### B. Résistance des contacts

La résistance des contacts est déterminée en utilisant la surface réelle des contacts.

$$R = \frac{l}{A} \quad (\text{A.8})$$

où :

$l$  : est l'épaisseur des contacts,

$A$  : est la surface réelle des contacts.

### C. Allure des arcs électriques [Fauchais 90] [Vacquié 96]

Les arcs électriques peuvent s'étudier, soit par la répartition des tensions, soit par la répartition de la densité des courants, soit par la répartition des champs électriques dans l'espace inter électrode (figure D-12). Au regard des phénomènes qui provoquent, les arcs électriques dans l'appareillage, il faut également envisager l'émission thermique. Sur la largeur de la surface des contacts, la température varie en fonction de la résistance de ces contacts.

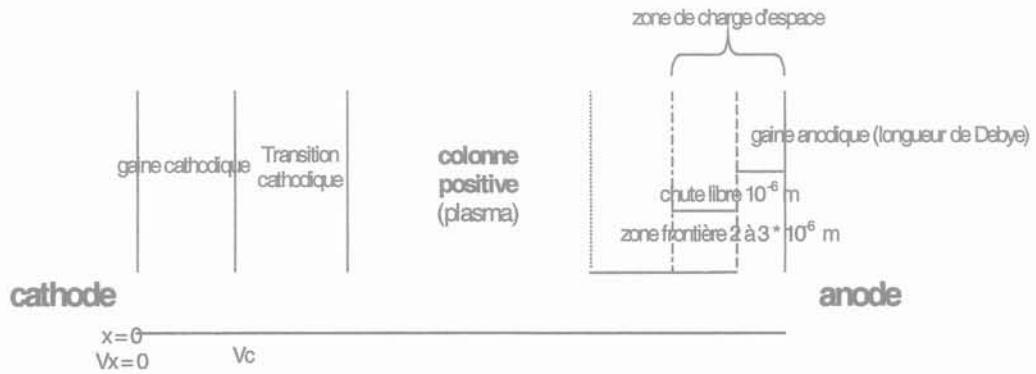


Figure D - 12 Zones dans l'espace inter électrode

En considérant la répartition de la densité des courants, les arcs électriques sont définis par les équations suivantes :

#### 1. Température

$$T = \frac{Ri}{\pi\sqrt{L}} \quad (\text{A.9})$$

où  $L$  : est le nombre de Lorenz =  $2,45 \cdot 10^{-8} \text{ V}^2/\text{K}^2$

#### 2. Conductivité électrique des gaz fortement ionisés

La conductivité électrique pour chaque point thermique sur la largeur de contact est basée sur le modèle de la colonne d'arc de Mayr :

$$\tau = \frac{26 \cdot 10^{-3}}{z} T^{3/2} \ln f \left( \frac{T}{p^{1/2}} \right) \quad (\text{A.10.1})$$

En supposant le nombre de charge égal à 1, l'équation (A.10) devient :

$$\tau = 15 \cdot 10^{-3} \cdot T^{3/2} \ln f \left( \frac{T}{p^{1/2}} \right) \quad (\text{A.10.2})$$

où :

$p$  : est la pression de gaz entre les contacts,

$z$  : est le nombre de charges,

$f$  : est la fréquence des courants conduits par les contacts.

### 3. Gaines anodique et cathodique

Les longueurs de la gaine anodique et de la gaine cathodique sont définies par la longueur de Debye :

$$\lambda = \left( \frac{kT_e \cdot y_o^{4/3}}{\frac{4}{9} V_o q^2} \right)^{3/4} \quad (\text{A.11})$$

où :

$k$  : est la constante de Boltzmann =  $1,38 \cdot 10^{-23}$  J/K,

$T_e$  : est la température sur la surface des contacts,

$V_o$  : est la tension appliquée sur la surface anodique,

$y_o$  : est la distance entre les contacts,

$q$  : est le nombre de charges.

### 4. Densité de courant

Dans la gaine anodique ou cathodique, la densité de courant est :

$$J = -\frac{4}{9} \frac{\varepsilon}{y^2} \left( \frac{-2qe}{m_e} \right)^{1/2} V^{3/2} \quad (\text{A.12})$$

La tension à une certaine distance de l'anode est :

$$V = V_o \left( \frac{y}{y_o} \right)^{4/3} \quad (\text{A.13})$$

Dans la colonne nous avons :

$$J = \tau E \quad (\text{A.14})$$

où le champ magnétique est :

$$E = -\frac{4}{3} \frac{V_o}{y_o} \left( \frac{y}{y_o} \right)^{1/3} \quad (\text{A.15})$$

$y$  : est une position entre l'anode et la cathode,

$e$  : est la charge d'un électron =  $-1,602 \cdot 10^{-19}$  C,

$\varepsilon$  : est la permittivité =  $\frac{1}{36\pi} 10^{-9}$  F/m,

$m_e$  : est la masse de l'électron =  $0.31 \cdot 10^{-31}$  kg,

## ANNEXE E :

## 1 . LES CATEGORIES DES OBJECTIFS D'EXPERIMENTATION [Picard 96][Pierre 96]

Catégories des dispositifs électriques	Facteurs considérés dans la conception	Paramètres d'entrée	Objectifs Intermédiaires	Réponses principales	Réponses additionnelles
Appareillage	force résiduelle (force non désirée développée par le flux résiduel qui dépend de l'intensité coercitive et de la perméance de l'entrefer)	l'entrefer minimum coercitive	mesure de force résiduelle	force résiduelle	flux rémanent champ d'excitation rémanent
	force (force dans l'entrefer nécessaire pour vaincre le ressort de rappel et la friction, et pour produire la vitesse désirée en considérant la pression de contacts)	raideur du ressort de rappel	mesure de force en fonction de la raideur du ressort de rappel	force	courant champ d'excitation pulsation de force (en courant alternatif) induction flux
		masse de la partie mobile grandeurs géométriques	mesure de force en fonction de la masse de la partie mobile ou des grandeurs géométriques de l'électro-aimant	force	
		courant d'alimentation	mesure de force en fonction du courant d'alimentation	force	
		inductance résistance de la bobine	mesure du temps de la réponse	temps de fermeture ou d'ouverture	il ne contient pas de courant comme la réponse  vitesse de déplacement temps d'établissement du courant force pulsation de force (en courant alternatif) courant de déclenchement ou d'encenchement courant de maintien
	amplitude max de la variation possible de l'entrefer (l'entrefer maximum nécessaire en considérant la compensation de courant conduit entre les contacts qui dépend du type de circuit de puissance connecté)	déflexion et tolérance de contacts type de charge	coupure	vitesse de rétablissement de la tension entre les bornes d'entrée et de sortie intensité du courant à couper	
		matériau des contacts	essais endurance mécanique	nombre minimal de manœuvres	
	efficacité mécanique				
	efficacité magnétique des tôles	courbe B(H) grandeurs géométriques	essais de propriété des tôles	pertes hystérésis courant Foucault	courant force
	chaleur: (l'augmentation de la température doit être raisonnable pour la durée de vie de l'isolation de la bobine)	coefficient température de résistance de la bobine  matériau des contacts  matériau des contacts	échauffement dus aux courants permanents  échauffement dus aux courants court circuit	température  température	Résistance Ro & Rt de la bobine pertes Joule

Catégories des dispositifs électriques	Facteurs considérés dans la conception	Paramètres d'entrée	Objectifs intermédiaires	Réponses principales (non complète)	Réponses additionnelles (non complète)
Transformateur	Isolément		essais de la tension à la fréquence industrielle (fI)		
			essais par tension induite à une fréquence > fI		
			essais de chocs électriques		
			mesures relatives aux décharges partielles		
	chaleur	coefficient température de résistance	échauffement dus aux courants permanents	température	résistances Ro & Rt pertes Joule
			échauffement dus aux courants de court-circuit	température	la tenue aux efforts électrodynamiques résistances Ro & Rt pertes Joule
	caractéristiques mécaniques		mesure de niveau du bruit		
			mesure d'intensité vibratoire		
			essais d'étanchéité		
			essais relatifs au degré de protection		
			essais des réservoirs sous pression		
			essais de chocs mécaniques		
	caractéristiques électromagnétiques	intensité coercitive	mesure des pertes magnétiques	pertes Foucault pertes hystérésis	courant magnétisant
			rendement	rendement	pertes Joule courant en court circuit pertes magnétiques
			mesure de rapport de transformation	rapport	tension primaire tension secondaire
			mesure des résistances de bobine	résistance primaire résistance secondaire	courant et tension primaire courant et tension secondaire



Catégories des dispositifs électriques	Facteurs considérés dans la conception	Paramètres d'entrée	Objectifs Intermédiaires	Réponses principales (non complète)	Réponses additionnelles (non complète)
Machine tournante	isolation		essais de la tension à la fréquence industrielle (fi)		
			essais par tension induite à une fréquence > fi		
	chaleur (l'augmentation de la température doit être raisonnable pour la durée de vie de l'isolation de la bobine)		échauffement dus aux courants permanents	température	pertes Joule résistances Ro & Rt
		coefficient température de résistance	échauffement dus aux courants de court-circuit	température	la tenue aux effort électrodynamiques résistances Ro & Rt pertes Joule
Alternateur			échauffement dus aux surintensités de courte durée	température	résistances Ro & Rt pertes Joule
	efficacité magnétique	surface des pôles et valeur de l'entrefer	rendement magnétique	rendement magnétique	pertes par hystérésis pertes par courants de Foucault pertes Joule sur bobinage primaire flux rémanent énergies stocké dans tôle et entrefer courant magnétisant
	caractéristiques mécaniques		mesure des pertes mécanique	rendement mécanique	énergie mécanique utilisée énergie mécanique produite vitesse couple
			couple en fonction de la vitesse	couple	vitesse couple max. puissance
			essais de chocs mécaniques		
			degré de protection procurés par les enveloppes		
			mesure d'intensité vibratoire		
			mesure du niveau de bruit		
moteurs		survitesse admissible	mesure de couple démarrage	couple démarrage	courant démarrage vitesse couple minimal
moteurs asynchrones			mesure de couple décrochage	couple décrochage	vitesse
machine asynchrone			mesure de glissement	glissement	vitesse
moteurs pas à pas			couple résiduel	couple résiduel	courant
			couple de maintien	couple maintien	courant

Catégories des dispositifs électriques	Facteurs considérés dans la conception	Paramètres d'entrée	Objectifs intermédiaires	Réponses principales (non complète)	Réponses additionnelles (non complète)
	caractéristiques électriques		facteur puissance en fonction de la puissance	facteur puissance	puissance
			le courant (fourni & absorbé) en fonction de la puissance	courant fourni ou absorbé	puissance
			rendement	rendement	pertes joules pertes magnétiques pertes mécaniques
			mesure des résistances et inductances	résistances inductances	tension courant
Génératrice			la variation de tension selon la charge	variation de tension	courant tension facteur puissance puissance
			irrégularité de la forme d'onde	harmonique	
Machine Synchrone			courbe d'excitation	courant magnétisant	puissance
			diverses réactances		
Alternateurs			mesure des courants de court circuit		
			contrôle de chutes de tension selon la charge		

## 2 . LES COMPORTEMENTS PHYSIQUES DU CONTACTEUR ET LES OBJECTIFS D'EXPERIMENTATION CORRESPONDANTS

Alimentation de la bobine	Comportements physiques		Facteurs influents	Algorithme du calcul de performances physiques	Hypothèse	Contraintes	Objectif d'expérimentation
	Position des pôles de contacteur	Performance observée					
courant alternatif courant continu	position ouverte (avant position repos)	force (d'appel) voltage (d'appel) courant (d'appel) courant de Foucault pertes Joule pertes hystérétiques (aimant non linéaire) effet des spires de Frager	* forme géométrique de circuit magnétique * dimension de circuit magnétique * masse de la partie mobile			<p><b>Géométrie</b> (la géométrie est choisie pour avoir un fonctionnement brusque à l'ouverture comme à la fermeture)</p> <p>Largeur entre branches doit être suffisamment grande par rapport à la course totale du contacteur afin d'éviter des fuites de flux magnétiques trop importantes entre les branches</p> <p><math>AFF &gt; CE + CO</math></p>	essais de propriété des tôles
	position fermée (avant position fermée)	force (de maintien) voltage (de maintien) courant (de maintien) courant de Foucault pertes Joule pertes hystérétiques (non linéaire) effet des spires de Frager	* effort résistant (raideur du ressort de rappel + compression initiale du ressort de rappel + raideur du ressort des pôles + compression initiale du ressort de pôles) * matériau des composants de contacteur			<p>Largeur de branche peut être inférieure au diamètre de rivet augmenté d'une largeur suffisante à la tenue mécanique de tôles</p> <p>Il existe une section minimale pour noyau en dessous de la quelle il n'est pas possible de construire l'électroaimant :  <math>2 \times \text{largeur de branche} \times \text{épaisseur}</math></p>	<p>essais de propriété des tôles</p> <p>échauffement dus aux courants permanents</p>



## ANNEXE F : CARACTERISTIQUES DES AGENTS INTERACTIFS DU LVEE

Agent	Fonctionnement
	<i>Agent parent représentant le système du Laboratoire Virtuel en Electrotechnique</i>
VELaboratory	<p>Cet agent contrôle les données pendant la simulation et maintient les fichiers des scénarios.</p> <p>Il contrôle aussi le processus de simulation numérique (le pilotage des logiciels de simulation).</p> <p>L'agent VELaboratory gère la distribution aux agents fils des résultats produits par un logiciel de simulation.</p> <p>La facette Présentation de cet agent visualise le monde virtuel de la salle d'expérimentation et le navigateur pour naviguer dans le monde virtuel.</p>
	<i>Agent fils représentant l'armoire des instruments utilisés dans l'expérimentation de dispositifs électriques</i>
Cupboard	<p>L'agent Cupboard organise les paramètres d'entrée contenant le type du dispositif électrique étudié, l'objectif de l'expérimentation et les conditions que le dispositif électrique doit vérifier.</p> <p>La facette Présentation de cet agent visualise un monde virtuel contenant les boîtes 3D en structure d'arbre et les menus dialogues.</p>
	<i>Agent fils qui organise les manipulations de l'expérimentation.</i>
Experiment	<p>Cet agent groupe des dispositifs électriques des sources électriques et des instruments de mesure dans le montage de l'expérimentation.</p>
	<i>Agent fils qui organise les catégories de dispositifs électriques.</i>
Device	<p>Cet agent gère et contrôle les catégories de dispositifs électriques.</p>
	<i>Agent fils représentant le contacteur qui est une catégorie de dispositifs électriques.</i>
Contacteur	<p>Cet agent organise et contrôle le type de contacteur choisi pour être étudié</p>
	<i>Agent fils représentant un contacteur avec circuit magnétique en forme de E.</i>
ContRelaiE	<p>Cet Agent définit le modèle de contacteur possédant un circuit magnétique en E : performances géométriques, comportements physiques (caractéristiques électromagnétiques, mécanique et thermique).</p> <p>L'agent ContRelaiE gère les informations (les paramètres d'entrée) qui décrivent la géométrie, les matériaux et les caractéristiques de fonctionnement.</p> <p>La facette Présentation de cet agent visualise le modèle virtuel du contacteur et anime les comportements physiques.</p> <p>La facette Abstraction contrôle les comportements physiques du contacteur pour chaque modèle physique</p>
ContRelaiP	<i>Agent fils représentant un contacteur avec un circuit magnétique avec armature plongeante.</i>
	<i>Agent fils représentant un contacteur avec un circuit magnétique en forme de U.</i>
ContRelaiU	
	<i>Agent fils représentant le transformateur qui est une catégorie de dispositifs électriques.</i>
Transformator	<p>Cet agent organise et contrôle le type de transformateur choisi pour être étudié.</p>
	<i>Agent fils représentant l'autotransformateur qui est un transformateur avec une seule bobine pour le côté primaire et le côté secondaire.</i>
AutoTransformator	<p>Cet agent définit le modèle de l'autotransformateur : performances géométriques, comportements physiques (caractéristiques électromagnétiques, mécanique et thermique).</p> <p>L'agent AutoTransformator gère les informations (les paramètres d'entrée) qui décrivent la géométrie, le matériau et la caractéristique de fonctionnement.</p> <p>La facette Présentation de cet agent visualise le modèle virtuel du transformateur et anime les comportements physiques.</p>
TransformatorTwo	<i>L'agent fils qui représente un transformateur possédant deux bobines une pour le côté primaire et l'autre pour le côté secondaire</i>
TransformatorThree	<i>Agent fils qui représente un transformateur possédant trois bobines, une pour le côté primaire, deux pour le côté secondaire</i>
	<i>Agent fils représentant la machine à courant continu qui est une catégorie de dispositifs électriques.</i>
MachineDC	<p>Cet agent organise et contrôle le type de machine à courant continu choisi pour être étudié.</p>

Agent	Fonctionnement
	<i>Agent fils représentant un générateur de courant continu.</i>
GeneratorDC	Cet agent définit le modèle de générateur à courant continu : performances géométriques et comportements physiques (caractéristiques électromagnétiques, mécanique et thermique). L'agent GeneratorDC gère les informations (les paramètres d'entrée) qui décrivent la géométrie, les matériaux et les caractéristiques de fonctionnement. La facette Présentation de cet agent visualise le modèle virtuel du générateur et anime les comportements physiques.
	<i>Agent fils représentant un moteur à courant continu.</i>
MotorDC	
	<i>Agent fils représentant un moteur courant à continu pas à pas.</i>
MotorStepDC	
	<i>Agent fils représentant une machine à courant alternatif qui est une catégorie de dispositifs électriques.</i>
MachineAC	Cet agent organise et contrôle le type de machine à courant alternatif choisi pour être étudié.
	<i>Agent fils représentant un générateur synchrone.</i>
GeneratorSynchrone	Cet agent définit le modèle de Générateur synchrone : performances géométriques, comportements physiques (caractéristiques électromagnétiques, mécanique et thermique). L'agent GeneratorSynchrone gère les informations (les paramètres d'entrée) qui décrivent la géométrie, les matériaux et les caractéristiques de fonctionnement. La facette Présentation de cet agent visualise le modèle virtuel du générateur et anime les comportements physiques.
	<i>Agent fils représentant un moteur synchrone.</i>
MotorSynchrone	
	<i>Agent fils représentant un moteur synchrone pas à pas</i>
MotorStepSynchrone	
	<i>Agent fils représentant un moteur asynchrone</i>
MotorAsynchrone	
	<i>Agent fils représentant un générateur asynchrone</i>
GeneratorAsynchrone	
	<i>Agent fils représentant un moteur asynchrone pas à pas</i>
MotorStepAsynchrone	
	<i>Agent fils qui organise les catégories de l'ensemble de sources électriques et des instruments de mesure.</i>
Montage	Cet agent gère et contrôle les objectifs d'expérimentation de contacteurs, de transformateurs et de machines électriques.
	<i>Agent fils qui organise les montages pour étudier les caractéristiques des contacteurs</i>
ContacteurMontage	Cet agent gère et contrôle l'ensemble des sources électriques et des instruments de mesure qui sera utilisé pour certains objectifs d'expérimentation de contacteur.

Agent	Fonctionnement
ContactorForceCurrentMontage	<p>Agent fils représentant le montage de contacteur utilisé pour étudier la force magnétique en fonction du courant de la bobine.</p> <p>Cet agent définit l'ensemble des sources électriques et des instruments de mesure utilisés pour mesurer la force et d'autres caractéristiques.</p> <p>L'agent ContactorForceCurrentMontage gère les informations (les paramètres d'entrée) qui décrivent le type de la source électrique et le niveau de méthode numérique qui sera utilisé pour le processus de la simulation numérique.</p> <p>La facette Présentation de cet agent visualise le modèle virtuel de la source électrique et des instruments de mesure : performances géométriques et performances de fonctionnement.</p>
ContCoupureMontage	<p>Agent fils représentant le montage de contacteur utilisé pour étudier la capacité de coupure des contacts.</p> <p>Cet agent définit l'ensemble de la source électrique et des instruments de mesure utilisés pour mesurer le nombre de coupures du contacteur pour une ouverture en court circuit ou sous tension.</p> <p>L'agent ContactorCoupureMontage gère les informations (les paramètres d'entrée) qui décrivent le type de la source électrique et le niveau de méthode numérique qui sera utilisé pour le processus de la simulation numérique.</p>
ContForceMasseMontage	<p>Agent fils représentant le montage de contacteur utilisé pour étudier la force en fonction de la masse de l'armature</p> <p>Cet agent définit l'ensemble de la source électrique et des instruments de mesure utilisés pour mesurer la force et d'autres caractéristiques.</p> <p>L'agent ContactorForceMasseMontage gère les informations sur la source électrique et la méthode d'analyse numérique.</p>
ContForceRappelMontage	<p>Agent fils représentant le montage de contacteur utilisé pour étudier la force en fonction du ressort de rappel</p>
ContForceResiduMontage	<p>Agent fils représentant le montage de contacteur utilisé pour étudier la force résiduelle due au phénomène d'hystérésis.</p>
ContHeatCourtMontage	<p>Agent fils représentant le montage de contacteur utilisé pour étudier les caractéristiques thermiques lorsque il y a un courant pendant un temps court</p>
ContHeatPermanentMontage	<p>Agent fils représentant le montage de contacteur utilisé pour étudier les caractéristiques thermiques pendant l'opération</p>
ContMagnetPropertyMontage	<p>Agent fils représentant le montage du contacteur utilisé pour étudier les propriétés du circuit magnétique.</p>
ContMechanicalEnduranceMontage	<p>Agent représentant le montage de contacteur utilisé pour étudier les pôles pendant la fermeture.</p>
ContTimeRespondMontage	<p>Agent fils représentant le montage de contacteur utilisé pour étudier la réponse du contacteur à la fermeture et à l'ouverture.</p>
TrafoMontage	<p>Agent fils qui organise les montages pour étudier les caractéristiques d'un transformateur</p> <p>Cet agent organise et contrôle l'ensemble des sources électriques et des instruments de mesure pour les expérimentations d'un transformateur.</p>
MachineMontage	<p>Agent fils qui organise les montages pour étudier les caractéristiques des machines électriques</p> <p>Cet agent organise et contrôle l'ensemble des sources et des instruments de mesure pour les expérimentations de machines électriques.</p>
ExperimentStandard	<p>Agent fils qui organise la norme d'expérimentation prise comme référence pour faire l'expérimentation du dispositif électrique.</p> <p>Cet agent maintient et organise le protocole d'expérimentation, hypothèses et contraintes pour certains objectifs d'expérimentation.</p> <p>La facette Présentation de cet agent visualise les textes de la norme d'expérimentation.</p>

## ANNEXE G :

## 1. LES CLASSES DE GESTION DES MATERIAUX

Classes	Propriétés	Fonctionnement
<p><i>Material</i></p> <p>note : le type de matériau est déterminé par la valeur de chaque propriété</p>	<ol style="list-style-type: none"> <li>1. resistivity(new Double) note : résistivité (ohm.m)</li> <li>2. resistivityCoef(new Double) note : coefficient résistivité (/°C)</li> <li>3. permeability(new Double) note : perméabilité (H/m)</li> <li>4. permitivity(new Double) note : permittivité (F/m)</li> <li>5. massDensity(new Double) note : densité masse (kg/m3)</li> <li>6. coercivity(new Double) note : coercivité (A/m)</li> <li>7. thermalConductivity(new Double) note : conductivité thermique</li> <li>8. thermalCapacity(new Double) note : capacité thermique</li> <li>9. tensileStrength(new Double) note : tension de solidité (kg/m<sup>2</sup>)</li> <li>10. hysteresisCharacteristic(new Double) note : caractéristique hystérésis (J/T<sup>2</sup>.kg)</li> <li>11. linearity(new Integer) note : linéarité matériau magnétique</li> <li>12. saturation(new Double) note : induction à saturation (T)</li> <li>13. slope(new Double) note : pente à l'origine de la courbe B(H)</li> <li>14. point(new Integer) note : nombre des points pour créer la courbe B(H)</li> <li>15. pointsH() note : retourner vecteur des champs magnétiques</li> <li>16. pointsB() note : retourner vecteur des inductions magnétiques</li> </ol>	<p>objet d'un matériau</p>

Classes	Propriétés	Fonctionnement
<i>MaterialDataCollection</i>	<ol style="list-style-type: none"> <li>1. <code>createMaterial(String nom)</code> note : créer et retourner un objet de matériau nommé <i>nom</i></li> <li>2. <code>addMaterial(Material material)</code> <code>addMaterials(Material[] materials)</code> <code>addMaterials(Enumeration materials)</code>  note : ajouter un matériau ou quelques matériaux dans un vecteur en vérifiant si le matériau déjà existe</li> <li>3. <code>contains(Material material)</code> note : vérifier si le matériau <i>avec certain nom</i> est déjà disponible</li> <li>4. <code>copyInto(Material[] materials)</code> note : copier les matériaux dans la table de matériaux</li> <li>5. <code>materialWhoseName(String nom)</code> note : retourner matériau nommé <i>nom</i></li> <li>6. <code>materialAt(int index)</code> note : retourner matériau se trouvant à l'index défini</li> <li>7. <code>materials()</code> note : retourner énumération des matériaux</li> <li>8. <code>material_names()</code> note : retourner énumération des noms de matériau</li> <li>9. <code>removeAllMaterials()</code> note : supprimer tous les matériaux</li> <li>10. <code>removeMaterial(Material material)</code> note : supprimer le matériau</li> <li>11. <code>size()</code> note : nombre de matériaux existant dans le vecteur</li> <li>12. <code>registerMaterial()</code> note : enregistrer les matériaux et ses propriétés dans un fichier texte, <i>Material.dat</i></li> </ol>	<ol style="list-style-type: none"> <li>1. la collection des matériaux</li> <li>2. créer un vecteur des objets matériau du fichier <i>Material.dat</i></li> <li>3. créer un objet matériau</li> <li>4. ajouter, vérifier ou supprimer un objet matériau dans le vecteur</li> <li>5. enregistrer le contenu du vecteur des matériaux dans le fichier <i>Material.dat</i></li> </ol>
<i>ListMaterial</i>	<ol style="list-style-type: none"> <li>1. <code>createMaterial(String nom)</code> note : créer et retourner un objet de matériau nommé <i>nom</i> et ajouter l'objet dans la collection des matériaux</li> <li>2. <code>namesOfMaterials()</code> note : récupérer un vecteur de nom des matériaux</li> <li>3. <code>getMaterial(String nom)</code> note : récupérer un matériau nommé <i>nom</i> de la collection des matériaux</li> <li>4. <code>deleteMaterial(String nom)</code> note : supprimer un matériau nommé <i>nom</i> de la collection des matériaux</li> </ol>	<ol style="list-style-type: none"> <li>1. la liste des matériaux</li> <li>2. créer la collection des matériaux</li> <li>3. récupérer les noms des matériaux</li> <li>4. gérer le contenu de la collection des matériaux</li> <li>5. redéfinir l'enregistrement de la collection des matériaux</li> </ol>



Classes	Propriétés	Fonctionnement
	5. registerMaterial() note : enregistrer les matériaux	
<i>CreationMaterialMenu</i>	contenir la liste des matériaux	Interface graphique pour créer un matériau et leurs propriétés
<i>DialogueCreationB_HCurve</i>	contenir la liste des matériaux	Interface graphique pour définir les points de la courbe B(H) si le matériau créé est magnétique non linéaire
<i>ListMaterialMenu</i>	contenir la liste des matériaux	Interface graphique pour voir la liste des matériaux et leurs propriétés

## 2. LES CLASSES DE GESTION DES NORMES D'EXPERIMENTATION

Classes	Propriétés	Fonctionnement
<p><i>ExperimentStandard</i></p>	<ol style="list-style-type: none"> <li>1. <code>setConstraints(Constraints <i>const</i>)</code> note : définit les contraintes utilisées pour obtenir les phénomènes se produisant</li> <li>2. <code>setExperimentProcedure(ExperimentProcedure <i>expProcec</i>)</code> note : définit le protocole d'expérimentation</li> <li>3. <code>setHypothesis(Hypothesis <i>hypothese</i>)</code> note : définit l'hypothèse utilisée pour modéliser les phénomènes se produisant pendant l'expérimentation</li> <li>4. <code>setLoad(Load <i>load</i>)</code> note : définit la charge pour le dispositif électrique étudié</li> <li>5. <code>setRegime(Regime <i>regime</i>)</code> note : définit le régime de fonctionnement du dispositif choisi pour atteindre l'objectif d'expérimentation</li> <li>6. <code>setNamePlate(NamePlate <i>plate</i>)</code> note : définit la plaque signalétique du dispositif étudié</li> </ol>	<ol style="list-style-type: none"> <li>1. Objet de la norme d'expérimentation</li> <li>2. Contenir les informations dont on a besoin pour faire une expérimentation de dispositif électrique</li> </ol>
<p><i>Constraints</i></p> <p>note : la classe parent des classes contraintes pour chaque catégorie de dispositif électrique</p>	<ol style="list-style-type: none"> <li>1. <code>toString()</code> note : retourner le nom de la contrainte</li> <li>2. <code>getContent()</code> note : retourner le contenu de la contrainte</li> <li>3. <code>keys()</code> note : retourner les mots clefs de la contrainte</li> <li>4. <code>elements()</code> note : retourner les valeurs de la contrainte</li> <li>5. <code>getLabels()</code> note : retourner les labels utilisés pour créer le contenu de la contrainte. Cette fonction est redéfinie par les sous classes</li> </ol>	<p>Contenir les contraintes que l'on s'impose durant la conception d'un dispositif électrique</p> <p>note : ces informations seront utilisées pour animer les performances physiques des dispositifs étudiés</p>
<ul style="list-style-type: none"> <li>• <i>ContactorConstraints</i></li> </ul>	<ol style="list-style-type: none"> <li>6. <code>temperatureMaximum(double <i>temperature</i>)</code> note : définir la température maximale qu'un contacteur peut supporter</li> <li>7. <code>closingVoltageMax(double <i>voltage</i>)</code> note : définir la tension maximale pour laquelle un contacteur doit réagir en fermeture</li> <li>8. <code>closingVoltageMin(double <i>voltage</i>)</code> note : définir la tension minimale pour laquelle un contacteur doit réagir en fermeture</li> <li>9. <code>openingVoltageMax(double <i>voltage</i>)</code> note : définir la tension maximale pour laquelle un contacteur doit réagir en ouverture</li> </ol>	

Classes	Propriétés	Fonctionnement
<ul style="list-style-type: none"> <li>• <i>TransfoConstraints</i></li> <li>• .....</li> </ul>	<ol style="list-style-type: none"> <li>10. <i>openingVoltageMin(double voltage)</i> note : définir la tension minimale pour laquelle un contacteur doit réagir en ouverture</li> <li>11. <i>findData(String key)</i> note : récupérer valeur de contrainte reliée au mot clef défini</li> <li>12. <i>wireConstraints()</i> note : définir le paramétrage des dimensions de la bobine pour chaque forme du circuit magnétique</li> <li>13. <i>shadeConstraints()</i> note : définir le paramétrage des dimensions des spires de Frager pour chaque forme du circuit magnétique</li> <li>14. <i>electromagnetConstraints()</i> note : définir le paramétrage de chaque forme du circuit magnétique</li> <li>15. <i>getGeoMessage()</i> note : retourner message à propos de la dimension</li> <li>16. <i>labelSize()</i> note : retourner la taille du contenu de contrainte</li> </ol>	
<i>Hypothesis</i>	<ol style="list-style-type: none"> <li>1. <i>toString()</i> note : retourner le nom de l'hypothèse</li> <li>2. <i>setContent(String hypothesis)</i> note : définir le contenu de l'hypothèse</li> <li>3. <i>getContent()</i> note : retourner le contenu de l'hypothèse</li> </ol>	Contenir les hypothèses utilisées pour exprimer les performances du dispositif électrique étudié
<i>ExperimentProcedure</i>	<ol style="list-style-type: none"> <li>1. <i>toString()</i> note : retourner le nom du protocole d'expérimentation</li> <li>2. <i>setProcedure(String procedure)</i> note : définir le contenu du protocole d'expérimentation</li> <li>3. <i>getProcedure()</i> note : retourner le contenu de protocole d'expérimentation</li> </ol>	Contenir les informations à propos des étapes qui doivent être faites pour atteindre un objectif d'expérimentation
<i>NamePlate</i>  note : la classe parent des classes NamePlate pour chaque catégorie de dispositif électrique	<ol style="list-style-type: none"> <li>1. <i>toString()</i> note : retourner le nom de la plaque signalétique</li> </ol>	Définir la plaque signalétique qui contient les paramètres de fonctionnement d'un dispositif électrique

Classes	Propriétés	Fonctionnement
	2. <code>setParameters(String parameter)</code> note : définir le contenu de la plaque signalétique  3. <code>getParameters()</code> note : retourner le contenu de la plaque signalétique  4. <code>resetParameters()</code> note : supprimer le contenu de la plaque signalétique  5. <code>removeParameter(String parameter)</code> note : supprimer un paramètre visualisé dans la plaque signalétique	
<ul style="list-style-type: none"> <li>• <i>ContactorNamePlate</i></li> </ul>	6. <code>setLabelParameter()</code> note : définir les paramètres de fonctionnement qui seront visualisés dans la plaque signalétique	
<ul style="list-style-type: none"> <li>• <i>TransfoNamePlate</i></li> <li>• .....</li> </ul>		
<b>Load</b>  note : <ul style="list-style-type: none"> <li>• ce type de charge exprime le couple parmi les valeurs associées</li> <li>• ce type de charge donne l'information pour traiter les valeurs associées</li> </ul>	1. types de charge : <ul style="list-style-type: none"> <li>• NO_LOAD = 1</li> <li>• SHORT_CIRCUIT = 2</li> <li>• RESISTIVE = 3</li> <li>• INDUCTIVE = 4</li> <li>• CAPACITIVE = 5</li> <li>• RESISTIVE_INDUCTIVE_PARALLELE = 6</li> <li>• RESISTIVE_INDUCTIVE_SERIE = 7</li> <li>• RESISTIVE_INDUCTIVE_CAPACITIVE_PARALLELE = 8</li> <li>• RESISTIVE_INDUCTIVE_CAPACITIVE_SERIE = 9</li> <li>• INDUCTIVE_CAPACITIVE_PARALLELE = 10</li> <li>• INDUCTIVE_CAPACITIVE_SERIE = 11</li> </ul> 2. <code>toString()</code> note : retourner le nom de la charge  3. <code>setLoadType(int typeOfLoad)</code> note : définir le type de la charge  4. <code>getLoadType()</code> note : retourner le type de la charge  5. <code>setCapacitance(double capacitance)</code> note : définir la valeur de la capacitance  6. <code>getCapacitance()</code> note : retourner la valeur de la capacitance  7. <code>setResistance(double resistance)</code> note : définir la valeur de la résistance	Définir le type de charge et les valeurs associées

Classes	Propriétés	Fonctionnement
	8. <code>getResistance()</code> note : retourner la valeur de la résistance  9. <code>setInductance(double inductance)</code> note : définir la valeur d'inductance  10. <code>getInductance()</code> note : retourner la valeur d'inductance	
<i>Regime</i>	1. type de régime : <ul style="list-style-type: none"> <li>• TRANSITORY = 1</li> <li>• STATIC = 2</li> <li>• SUBTRANSITORY = 3</li> </ul> 2. <code>toString()</code> note : retourner le nom du régime  3. <code>setRegime( int regime)</code> note : définir le type de régime  4. <code>getRegime()</code> note : retourner le type de régime	Définir les conditions de travail d'un dispositif électrique pour certain atteindre un objectif d'expérimentation
<i>CollectionExperimentStandard</i>	1. <code>createExperimentStandard(String nom)</code> note : créer et retourner un objet de la norme d'expérimentation nommé nom  2. <code>addExperimentStandard(ExperimentStandard expstandard)</code>  <code>addExperimentStandards(ExperimentStandard[] expstandards)</code>  <code>addExperimentStandards(Enumeration expstandards)</code> note : ajouter une ou quelques normes d'expérimentation en vérifiant si la norme existe déjà  3. <code>contains(ExperimentStandard expstandard)</code> note : vérifier si une norme d'expérimentation avec un certain nom déjà existe  4. <code>copyInto(ExperimentStandard[] expstandards)</code> note : copier les normes d'expérimentations définies dans la table  5. <code>expStandardWhoseName(String nom)</code> note : retourner une norme d'expérimentation nommée nom  6. <code>expStandardAt(int index)</code> note : retourner une norme d'expérimentation se trouve à l'index défini  7. <code>expStandard_names()</code> note : retourner une énumération des noms de la norme d'expérimentation  8. <code>expStandards()</code> note : retourner une énumération des normes d'expérimentation	1. la collection des normes d'expérimentation  2. créer un vecteur des objets de la norme d'expérimentation qui s'enregistre dans le fichier ExpStandard.dat  3. créer un objet de la norme d'expérimentation  4. ajouter, vérifier ou supprimer un objet de la norme d'expérimentation  5. enregistrer le contenu du vecteur des normes d'expérimentation dans le fichier ExpStandard.dat

Classes	Propriétés	Fonctionnement
	<p>9. <code>removeAllExperimentStandards()</code> note : supprimer toutes les normes d'expérimentation contenues</p> <p>10. <code>removeExperimentStandard(ExperimentStandard <i>expstandard</i>)</code> note : supprimer la norme d'expérimentation définie</p> <p>11. <code>registerExperimentStandard()</code> note : enregistrer les normes d'expérimentation dans un fichier texte, <code>ExpStandard.dat</code></p>	
<i>ListExperimentStandard</i>	<p>1. <code>createExperimentStandard(String <i>nom</i>)</code> note : créer et retourner un objet de la norme d'expérimentation et ajouter l'objet dans la collection des normes d'expérimentation</p> <p>2. <code>namesOfExperimentStandards()</code> note : retourner un vecteur des noms de la norme d'expérimentation</p> <p>3. <code>getExperimentStandard(String <i>nom</i>)</code> note : retourner une norme d'expérimentation nommée <i>nom</i></p> <p>4. <code>deleteExperimentStandard(String <i>nom</i>)</code> note : supprimer une norme d'expérimentation nommée <i>nom</i></p> <p>5. <code>registerExperimentStandard()</code> note : enregistrer les normes d'expérimentation</p>	<p>1. la liste des normes d'expérimentation</p> <p>2. créer la collection des normes d'expérimentation</p> <p>3. récupérer les noms de la norme d'expérimentation</p> <p>4. gérer le contenu de la collection des normes d'expérimentation</p> <p>5. redéfinir l'enregistrement de la collection des normes d'expérimentation</p>
<i>ExperimentStandardCreationMenu</i>	Contenir la liste des normes d'expérimentation	interface graphique pour créer une norme d'expérimentation les informations associées

## ANNEXE H : LE PILOTAGE DE LOGICIELS DE SIMULATION

Système de simulation	Interface de l'outil de communication	Outil de communication
<p style="text-align: center;"><b>ElementsRoters</b></p> <p>(le système de simulation numérique basé sur les composants de Roters : utilisation de coefficient de fuite pour calculer le flux magnétique)</p> <p><b>par l'interface graphique</b></p> <ol style="list-style-type: none"> <li>créer un projet et lancer le module de calcul <p><i>production</i> : un fichier PROCEED contenant les paramètres et les phrases de commande</p> </li> <li>sauvegarder le projet <p><i>production</i> : un fichier &lt;nom_projet&gt; contenant les valeurs des paramètres et les résultats de la simulation numérique</p> </li> <li>via menu « open file », recréer un projet existant</li> </ol> <hr/> <p><b>par pilote</b></p> <ol style="list-style-type: none"> <li>lancer ElementsRoters en utilisant le fichier PROCEED <p><i>production</i> :</p> <ul style="list-style-type: none"> <li>un fichier PROCEED.txt</li> <li>un fichier RESULT.txt qui contient les données comme dans le fichier &lt;nom_projet&gt;</li> </ul> </li> </ol>	<ol style="list-style-type: none"> <li>Créer le fichier PROCEED des fichiers : <ul style="list-style-type: none"> <li>PROCEED_U pour circuit magnétique en U</li> <li>PROCEED_E pour circuit magnétique en E</li> <li>PROCEED_I pour circuit magnétique en I</li> <li>PROCEED_O pour circuit magnétique en O</li> </ul> <p>en transférant les données créées pendant la création de scénario de l'expérimentation dans Laboratoire Virtuel en Génie Electrique (LVEE)</p> </li> </ol> <hr/> <ol style="list-style-type: none"> <li>lancer pilotage de ElementsRoters</li> </ol> <hr/> <ol style="list-style-type: none"> <li>Récupérer l'induction et le champ magnétique (du fichier RESULT.txt)</li> </ol>	<p style="text-align: center;"><i><b>pilote de ElementsRoters</b></i></p>

Système de simulation	Interface de l'outil de communication	Outil de communication
<p><b>Flux2D</b></p> <p>(le système de simulation numérique basé sur la méthode des éléments finis en 2D)</p> <p><b>par l'interface graphique</b></p> <ol style="list-style-type: none"> <li>créer la géométrie et le maillage (avec programme preflu) <p><i>production :</i></p> <ul style="list-style-type: none"> <li>un fichier preflu.esp contenant les commandes du programme preflu</li> <li>un fichier &lt;nom_géométrie&gt;.tra (une géométrie)</li> <li>un fichier &lt;nom_géométrie&gt;.lst contenant les paramètres géométriques créés</li> </ul> </li> <li>si nécessaire, créer le matériau (avec programme cslmat) <p><i>production :</i></p> <ul style="list-style-type: none"> <li>un fichier cslmat.esp contenant les commandes du programme cslmat</li> <li>un fichier Mat.dat comme base de données des matériaux</li> <li>un fichier Mat.lst contenant les matériaux créés</li> </ul> </li> <li>si nécessaire, créer le circuit électrique (avec programme circuit) <p><i>production :</i></p> <ul style="list-style-type: none"> <li>un fichier circuit.esp contenant les commandes du programme cslmat</li> <li>un fichier &lt;nom_circuit&gt;.tra (un circuit électrique)</li> </ul> </li> <li>créer les propriétés de la géométrie (avec programme prophy) <p><i>production :</i></p> <ul style="list-style-type: none"> <li>un fichier prophy.esp contenant les commandes du programme prophy</li> <li>un fichier &lt;nom_projet&gt;.tra</li> </ul> </li> </ol>	<ol style="list-style-type: none"> <li>recupérer les données d'un projet (par l'outil de communication) <p>note :</p> <ul style="list-style-type: none"> <li>l'interface détermine le projet qui sera utilisé</li> </ul> </li> </ol> <hr/> <ol style="list-style-type: none"> <li>modifier les paramètres géométriques</li> <li>modifier les propriétés <p>note : pour l'instant l'outil de communication ne tient pas de compte de la modification des propriétés de circuit externe</p> </li> <li>si le nom de matériau créé dans LVEE est différent de celui créé dans Flux2D, créer nouveau objet de matériau</li> </ol>	<ol style="list-style-type: none"> <li>recupérer les données d'un projet <p><b>paramètres géométriques</b></p> <ol style="list-style-type: none"> <li>créer un fichier espion, preflu.esp, contenant les commandes pour imprimer les paramètres géométriques</li> <li>lancer le programme preflu par pilotage</li> <li>recupérer les paramètres géométriques de fichier &lt;nom_géométrie&gt;.lst, en créant les objets des paramètres géométriques</li> </ol> <p><b>paramètres physiques</b></p> <ol style="list-style-type: none"> <li>créer un fichier espion, expgen.esp, contenant les commandes pour lire les propriétés d'une géométrie (les régions, les matériaux des régions, et les sources des régions)</li> <li>lancer le programme expgen par pilotage</li> <li>créer les objets des propriétés</li> </ol> <p><b>matériaux</b></p> <ol style="list-style-type: none"> <li>créer un fichier espion, cslmat.esp, contenant les commandes pour imprimer les matériaux créés</li> <li>lancer le programme cslmat par pilotage</li> <li>créer les objets des matériaux</li> </ol> <p>note : les objets des matériaux sont de la classe GenericMaterial. Ils ne définissent que le nom de matériau</p> </li> </ol> <hr/>

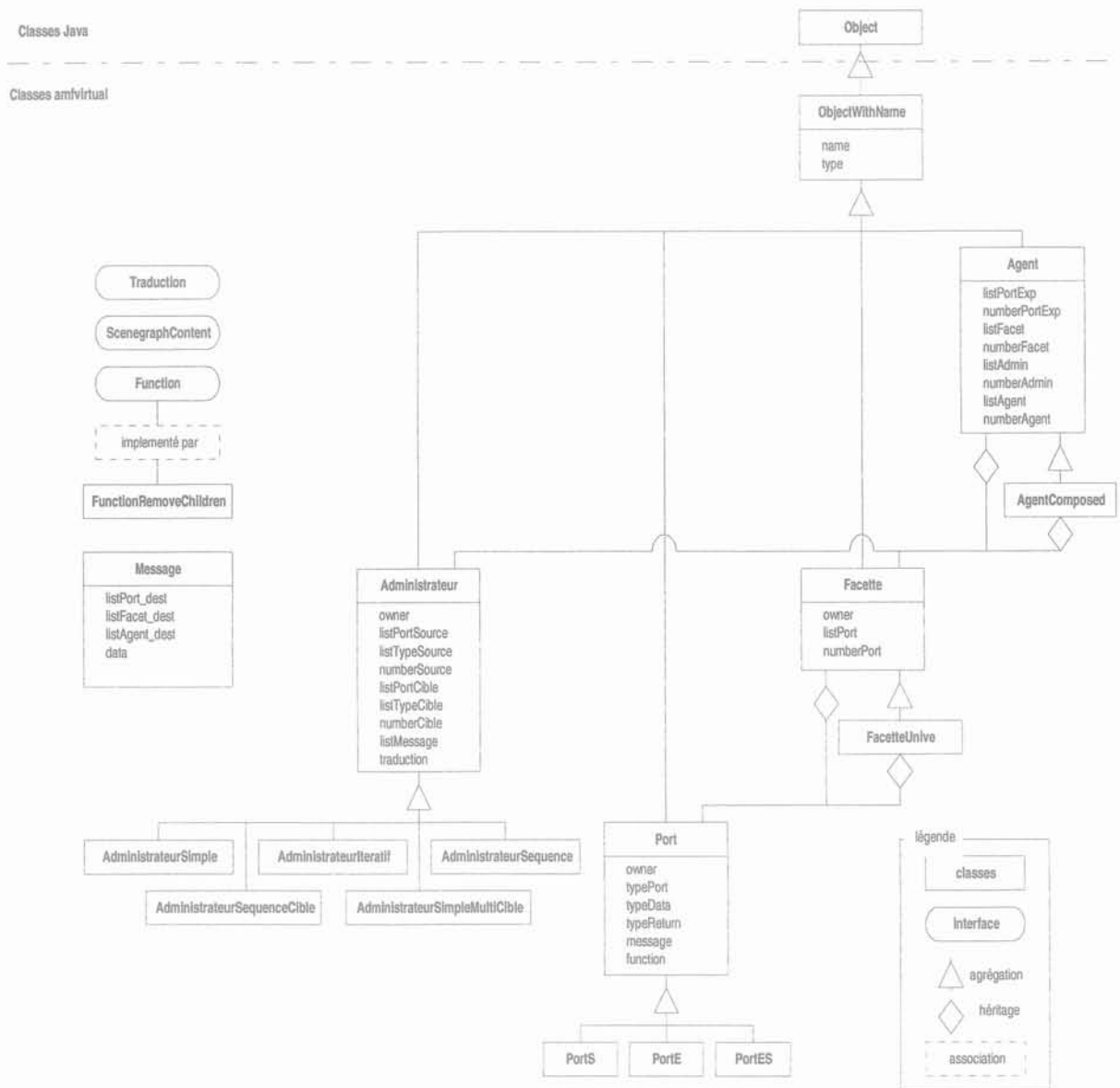


Système de simulation	Interface de l'outil de communication	Outil de communication
<p>5. lancer la simulation numérique (avec programme resgen)</p> <p><i>production :</i></p> <ul style="list-style-type: none"> <li>un fichier resgen.esp contenant les commandes du programme resgen</li> </ul> <p>6. exploiter les résultats (avec programme expgen)</p> <p><i>production :</i></p> <ul style="list-style-type: none"> <li>un fichier expgen.esp contenant les commandes du programme expgen</li> </ul> <hr/> <p><b>par pilote</b></p> <p>utiliser les fichiers espions :</p> <ol style="list-style-type: none"> <li>1. preflu.esp pour créer ou modifier les paramètres géométriques et remailler</li> <li>2. csimat.esp pour créer, ajouter ou modifier un matériau</li> <li>3. prophy.esp ou modpro.esp pour créer ou modifier les propriétés d'une géométrie</li> <li>4. resgen.esp pour lancer le calcul</li> <li>5. expgen.esp pour exploiter les résultats</li> </ol>	<p>5. lorsque le matériau magnétique est non linéaire, créer les données de calcul</p> <p>6. créer les résultats nécessités (et mettre les résultats dans liste de l'outil de communication)</p> <p>note : l'interface détermine le type des résultats</p> <hr/> <p>7. lancer calcul avec Flux2D</p> <hr/> <p>8. récupérer les valeurs des résultats</p>	<p>2. lancer Flux2D par pilotage</p> <ol style="list-style-type: none"> <li>a. si la liste de paramètres géométriques modifiés &gt; 0 : <ul style="list-style-type: none"> <li>créer preflu.esp</li> <li>lancer programme preflu</li> </ul> </li> <li>b. si la liste de nouveau matériau &gt; 0 : <ul style="list-style-type: none"> <li>créer csimat.esp</li> <li>lancer programme csimat</li> </ul> </li> <li>c. si la géométrie est modifiée : <ul style="list-style-type: none"> <li>créer prophy.esp</li> <li>lancer programme prophy</li> </ul> </li> <li>d. si la liste de propriétés modifiées &gt; 0 : <ul style="list-style-type: none"> <li>créer modpro.esp</li> <li>lancer programme prophy</li> </ul> </li> <li>e. créer resgen.esp et ensuite lancer programme resgen</li> <li>f. créer expgen.esp et ensuite lancer programme expgen</li> <li>g. remplir les valeurs des résultats concernés</li> </ol>

Système de simulation	Interface de l'outil de communication	Outil de communication
<p><b>Flux3D</b> (le système de simulation numérique basé sur la méthode des éléments finis en 3D)</p> <p><b>par l'interface graphique</b></p> <ol style="list-style-type: none"> <li>Créer entité géométrique : <ul style="list-style-type: none"> <li>boite infini</li> <li>point</li> <li>ligne</li> <li>surface</li> <li>volume</li> </ul> </li> <li>Définir paramètres physiques: <ul style="list-style-type: none"> <li>type d'application</li> <li>méthode de calcul</li> <li>caractéristique physique : <ul style="list-style-type: none"> <li>les propriétés (magnétique, électrique, thermique)</li> <li>les sources du dispositif étudié (ex : en magnétisme : courant électrique ou d'aimant, en électrostatique : densité de charge, en thermique : densité source thermique)</li> <li>la condition aux limites aux frontières du domaine modélisé (ex : potentiel électrique fixe en électrocinétique)</li> <li>la signification physique des limites géométriques du domaine (ex : infini, plan symétrique)</li> </ul> </li> </ul> </li> <li>Mailler</li> <li>Résoudre les équations du dispositif étudié par les algorithmes</li> <li>Interpréter les résultats de calcul des champs : <ul style="list-style-type: none"> <li>sous forme d'image</li> <li>sous forme numérique</li> </ul> </li> </ol> <p><b>par pilote</b></p> <p>utiliser le fichier espion</p>	<ol style="list-style-type: none"> <li>récupérer les données d'un projet (par l'outil de communication) <p>note :</p> <ul style="list-style-type: none"> <li>l'interface détermine le projet qui sera utilisé</li> </ul> </li> <li>modifier les paramètres géométriques</li> <li>modifier les propriétés</li> <li>si le nom de matériau créé dans LVEE est différent que celui créé dans Flux2D, créer nouvel objet matériau</li> <li>lancer calcul avec Flux3D</li> <li>récupérer les valeurs des résultats</li> </ol>	<ol style="list-style-type: none"> <li>créer un fichier espion et lancer Flux 3D par pilotage pour récupérer les données d'un projet <ul style="list-style-type: none"> <li>paramètres géométriques</li> <li>propriétés physiques (région, matériau et source)</li> <li>résultats interprétés</li> </ul> </li> <li>lancer Flux3D par pilotage <ol style="list-style-type: none"> <li>si la liste de paramètres géométriques modifiés &gt; 0 : <ul style="list-style-type: none"> <li>ajouter fichier espion avec les commandes pour modifier les paramètres géométriques</li> </ul> </li> <li>si la liste de nouveaux matériaux &gt; 0 : <ul style="list-style-type: none"> <li>ajouter fichier espion avec les commandes pour créer les matériaux</li> </ul> </li> <li>si la liste de propriétés modifiées &gt; 0 : <ul style="list-style-type: none"> <li>ajouter fichier espion avec les commandes pour modifier les propriétés physiques</li> </ul> </li> <li>lancer Flux3D en utilisant le fichier espion créé</li> <li>récupérer les nouvelles valeurs des résultats</li> </ol> </li> </ol>

## ANNEXE I :

## 1. MODELE OMT DES OBJETS AMF



## 2. LES CLASSES DES COMPOSANTS AMF

Classes	Propriétés	Fonctionnement (à propos de communication)
<p><i>ObjectWithName</i></p> <p>note : classe parent des composants AMF</p>	<ol style="list-style-type: none"> <li>type : <ul style="list-style-type: none"> <li>PORT = « port »</li> <li>AGENT = « agent »</li> <li>FACETTE = « facette »</li> <li>ADMINISTRATEUR = « administrateur »</li> </ul> <p>note : type des composants AMF</p> </li> <li>getName() note : retourner le nom de composant</li> <li>getType() note : retourner le type de composant</li> </ol>	
<p><i>Agent</i></p> <p>note : contient la définition de la facette Contrôle</p>	<ol style="list-style-type: none"> <li>addPortExportable(Port <i>port</i>) note : ajouter un port exportable (port qui peut s'activer ou être activé par l'administrateur d'un autre agent)</li> <li>deletePortExportable(Port <i>port</i>) note : supprimer un port exportable</li> <li>getPortExportable(String <i>nom</i>) note : retourner un port exportable avec le nom défini</li> <li>getPortExportables() note : retourner énumération des ports exportables</li> <li>addFacette(Facette <i>facette</i>) note : ajouter une facette</li> <li>deleteFacette(Facette <i>facette</i>) note : supprimer une facette</li> <li>getFacet(String <i>nom</i>) note : retourner une facette avec le nom défini</li> <li>getFacets() note : retourner énumération des facettes</li> <li>addAdministrator(Administrateur <i>admin</i>) note : ajouter un administrateur</li> <li>deleteAdministrator(Administrateur <i>admin</i>) note : supprimer un administrateur</li> <li>getAdministrator(String <i>nom</i>) note : retourner un administrateur avec le nom défini</li> </ol>	<ol style="list-style-type: none"> <li>recevoir message de facette (Abstraction ou Présentation)</li> <li>activer administrateur ayant de port actif</li> <li>envoyer message soit à facette (Abstraction ou Présentation) soit à l'autre agent</li> </ol> <pre> receive(message) { - vérifier s'il y a de port exportable actif - si le port exportable actif est port source (processus pour port source exportable) :     - si cet agent dans la liste des destinations de message, activeAdmins()     - envoyer message à l'autre agent - si le port exportable actif est port cible (processus pour port cible exportable) :     - si cet agent contient facette du port cible, <b>envoyer message à la facette</b>, si non <b>envoyer message à l'autre agent</b> - si le port exportable actif est port cible- source :     - si en cours il se fonctionne comme port cible, faire le processus pour port cible exportable     - si en cours il se fonctionne comme port source, faire le processus pour port source exportable - si aucun port exportable est actif, activeAdmins() } </pre>

Classes	Propriétés	Fonctionnement (à propos de communication)
	<p>12. <code>getAdministrators()</code> note : retourner énumération des administrateurs</p> <p>13. <code>getAgent(String nom)</code> note : retourner agent parent ou agent fils avec le nom défini</p> <p>14. <code>activeAdmins()</code> note : activer les administrateurs qui ont un port actif</p> <p>15. <code>receive(Message msg)</code> note : recevoir un message et déterminer si le message était envoyé à une facette ou à un autre agent</p> <p>16. <code>sendMessage(Message msg, String nom_agent, String nom_facette)</code> note : envoyer un message à un autre agent ou à une facette</p> <p>17. <code>avertirCible(Message msg)</code> note : avertir les facettes et/ou les agents ayant des ports cibles qui sont destination du message défini</p>	<p>4. avertir les agents ou les facettes qui contiennent les ports destinations de message</p> <p>note : fait via <code>Administrateur.traduction.translate()</code></p> <p><b>avertirCible(message) {</b></p> <ul style="list-style-type: none"> <li>- vérifier s'il y a de port exportable actif</li> <li>- si le port exportable actif se fonctionne <b>comme port cible</b> <ul style="list-style-type: none"> <li>▪ si le port n'appartient pas de cet agent, envoyer message à l'autre agent</li> <li>▪ si le port appartient de cet agent, envoyer message à facette</li> </ul> </li> <li>- s'il n'y a pas de port exportable actif, envoyer message à facette</li> </ul> <p><b>}</b></p>
<p>• <i>AgentComposed</i></p> <p>note : un agent composé fonctionne comme agent fils</p>	<p>18. <code>addAgent(Agent agent)</code> note : ajouter un agent fils</p> <p>19. <code>deleteAgent(Agent agent)</code> note : supprimer un agent fils</p> <p>20. <code>getAgentChildren()</code> note : retourner énumération des agents fils</p>	
<i>Facette</i>	<p>1. <code>setMessage(Message msg)</code> note : définir un message qui sera transféré par un port source</p> <p>2. <code>addPort(Port port)</code> note : ajouter un port, soit un port source ou un port cible et soit exportable ou non exportable</p> <p>3. <code>getPort(String nom)</code> note : retourner un port</p> <p>4. <code>getPorts()</code> note : retourner énumération des ports</p> <p>5. <code>deletePort(Port port)</code> note : supprimer un port</p> <p>6. <code>active(String nom_port)</code> note : activer un port source</p>	<p>1. activer port source</p> <p><b>active(nom_port) {</b></p> <ul style="list-style-type: none"> <li>- <code>portS = getPort(nom_port)</code></li> <li>- <code>portS.setMessage(message)</code></li> <li>- <code>portS.active()</code></li> </ul> <p><b>}</b></p>

Classes	Propriétés	Fonctionnement (à propos de communication)
<ul style="list-style-type: none"> <li><i>FacetteUnive</i>  note : classe héritage de classe Facette pour définir une facette Présentation qui contient un monde virtuel</li> </ul>	<ol style="list-style-type: none"> <li><code>send(Message msg)</code> note : envoyer un message à agent (facette contrôle)</li> <li><code>receive(Message msg)</code> note : recevoir un message et activer port destination du message</li> </ol> <hr/> <ol style="list-style-type: none"> <li><code>universe()</code> note : créer un monde virtuel (SimpleUniverse) où les objets virtuels seront placés</li> <li><code>getTG_VPUniverse()</code> note : retourner l'orientation du point de vue dans le monde virtuel</li> <li><code>getUniverse()</code> note : retourner le monde virtuel</li> <li><code>getCanvas()</code> note : retourner canvas3D qui contient le monde virtuel</li> <li><code>getVP_Position()</code> note : retourner la position du point de vue</li> <li><code>getDirectionalLight()</code> note : retourner la lumière qui se place dans le monde virtuel</li> </ol>	<ol style="list-style-type: none"> <li>recevoir message de facette Contrôle (agent) pour activer port cible  <code>receive(message) {</code>  <ul style="list-style-type: none"> <li>- vérifier s'il y a de port cible actif</li> <li>- si le port cible actif est destination de message : <ul style="list-style-type: none"> <li>▪ <code>supprimer le port cible de la liste des destinations de message</code></li> <li>▪ <code>portE.setMessage(message)</code></li> <li>▪ <code>portE.active()</code></li> </ul> </li> </ul> <code>}</code> </li> <li>envoyer message à facette Contrôle (agent)  note : fait via <code>PortS.active()</code>  <code>send(message) {</code>  <code>Agent.receive(message)</code>  <code>}</code> </li> </ol>
<i>Administrateur</i>	<ol style="list-style-type: none"> <li><code>addSource(String nom_port, String nom_facette, String nom_agent)</code> note : ajouter un port source avec le nom du port, le nom de facette où le port se trouve et le nom d'agent qui contient la facette</li> <li><code>deleteSource(String nom_port)</code> note : supprimer un port source avec le nom défini</li> <li><code>addCible(String nom_port, String nom_facette, String nom_agent)</code> note : ajouter un port cible avec le nom du port, le nom de facette où le port se trouve et le nom d'agent qui contient la facette</li> <li><code>deleteCible(String nom_port)</code> note : supprimer une cible avec le nom défini</li> <li><code>setMessage(String nom_port_source, Message msg)</code> note : définir un message avec le nom de port source qui va le transférer</li> </ol>	<ol style="list-style-type: none"> <li>recupérer nom de port actif</li> <li>recupérer message du port actif</li> <li>activer les ports cibles si le message vient de port source ou activer les ports source si le message vient de port cible</li> <li>rendre inactifs les ports d'où le message vient</li> <li>traduire le message</li> <li>libérer la liste de messages</li> <li>libérer l'étalage de l'ordre d'activation des ports</li> </ol>

Classes	Propriétés	Fonctionnement (à propos de communication)
	<p>6. <code>clearListMessage()</code> note : libérer la liste de messages</p> <p>7. <code>getMessages()</code> note : retourner énumération de messages</p> <p>8. <code>getMessage(String nom_port_source)</code> note : retourner un message qui est transféré par le port source défini</p> <p>9. <code>active(Message msg)</code> note : activer les ports cibles. Cette fonction sera redéfinie par les classes héritages</p> <p>10. <code>sourceActive()</code> note : vérifier si un port source est actif</p> <p>11. <code>getOrdreActive()</code> note : retourner table de l'ordre d'activation des ports sources</p> <p>12. <code>clearOrdreActive()</code> note : libérer table de l'ordre d'activation des ports sources</p> <p>13. <code>getNumberPortActive()</code> note : retourner nombre des ports sources qui sont actifs</p> <p>14. <code>getNameActivePort()</code> note : retourner les noms des ports sources actifs</p> <p>15. <code>cibleActive()</code> note : vérifier si un port cible est actif</p> <p>16. <code>numberOfSource()</code> note : retourner nombre des ports sources</p> <p>17. <code>numberOfCible()</code> note : retourner nombre des ports cibles</p> <p>18. <code>getPortsSource()</code> note : retourner énumération des ports sources</p> <p>19. <code>getPortsCible()</code> note : retourner énumération des ports cibles</p> <p>20. <code>getPortSource(String nom)</code> note : retourner un port source avec le nom défini</p> <p>21. <code>getPortCible(String nom)</code> note : retourner un port cible avec le nom défini</p>	<pre> <b>active(message) {</b>   - enum = getPortsSource()     nom_port = portActive.getName()    - setMessage(nom_port, message)    - enum = getPortsCible()     portCible.setIsActive(true)    - port =     getPortSource(getNameActivePort())     port.setIsActive(false)    - clearListMessage()    - clearOrdreActive()  <b>}</b> </pre>

Classes	Propriétés	Fonctionnement (à propos de communication)
	<p>22. <code>setTraduction(Traduction translation)</code> note : définir la fonction qui traite le message avant qu'il soit transféré au port cible</p> <p>23. <code>retourEnable(boolean retour)</code> note : définir si l'administrateur est capable de retourner un message du port cible au port source</p> <p>24. <code>getValuelsRetour()</code> note : savoir si l'administrateur est capable de retourner un message</p> <p>25. <code>getAgent()</code> note : retourner l'agent qui contient cet administrateur</p> <p>26. <code>getTurn()</code> note : retourner le tour de l'administrateur à propos de l'envoi de message (en transférant ou en retournant un message)</p>	
<ul style="list-style-type: none"> <li>• <i>AdministrateurIteratif</i> note : <ul style="list-style-type: none"> <li>• relier n ports sources avec un port cible</li> <li>• cet administrateur ne tient pas compte du mécanisme de retour</li> </ul> </li> </ul>	<p>Redéfinir la fonction active(Message <i>msg</i>)</p> <p>note : activer port cible une fois que tous les ports sources sont actifs. L'activation des ports cibles suit un certain ordre</p>	
<ul style="list-style-type: none"> <li>• <i>AdministrateurSequence</i> note : <ul style="list-style-type: none"> <li>• relier n ports sources avec un port cible</li> <li>• cet administrateur ne tient pas compte du mécanisme de retour</li> </ul> </li> </ul>	<p>Redéfinir la fonction active(Message <i>msg</i>)</p> <p>note : activer port cible une fois que tous les ports sources ont été activés en séquence</p>	
<ul style="list-style-type: none"> <li>• <i>AdministrateurSequenceCible</i> note : <ul style="list-style-type: none"> <li>• relier un port source avec n ports cibles</li> <li>• cet administrateur ne tient pas compte du mécanisme de retour</li> <li>• les ports cibles doivent être définis dans l'ordre de leur future activation</li> </ul> </li> </ul>	<p>Redéfinir la fonction active(Message <i>msg</i>)</p> <p>note : activer les ports cibles en séquence (à partir de premier port jusqu'à dernier port) dès qu'un port source est actif</p>	



Classes	Propriétés	Fonctionnement (à propos de communication)
<ul style="list-style-type: none"> <li><i>AdministrateurSimple</i> note :               <ul style="list-style-type: none"> <li>relier un port source ou n ports sources avec un port cible</li> <li>en cas liaison d'un port source avec un port cible, cet administrateur tient compte du mécanisme de retour</li> </ul> </li> </ul> <hr/> <ul style="list-style-type: none"> <li><i>AdministrateurSimpleMultiCible</i> note :               <ul style="list-style-type: none"> <li>relier n ports sources avec n ports cibles</li> <li>les noms des ports sources sont identiques aux noms des ports cibles</li> <li>cet administrateur ne tient pas compte du mécanisme de retour</li> </ul> </li> </ul>	<p>Redéfinir la fonction active(Message <i>msg</i>)</p> <p>note : activer port cible chaque fois qu'un port source est activé</p> <hr/> <p>Redéfinir la fonction active(Message <i>msg</i>)</p> <p>note : activer port cible ayant un nom identique au nom de port source actif</p>	
<p><i>Port</i></p>	<p>1. type de port :</p> <ul style="list-style-type: none"> <li>INPUT = 1</li> <li>OUTPUT = 2</li> <li>INPUT_OUTPUT = 3</li> </ul> <p>note : définir fonctionnement de port soit port cible(PortE), port source (PortS) ou port cible-source (PortES)</p> <p>2. type de donnée</p> <ul style="list-style-type: none"> <li>DOUBLE = "double"</li> <li>INTEGER = "integer"</li> <li>LONG = "long"</li> <li>SHORT = "short"</li> <li>STRING = "string"</li> <li>BOOLEAN = "Boolean"</li> <li>VECTOR = "vector"</li> <li>HYPOTHESIS = "hypothesis"</li> <li>CONSTRAINTS = "constraints"</li> <li>LOAD = "load"</li> <li>NAMEPLATE = "nameplate"</li> <li>EXPERIMENTPROCEDURE = "expProcedure"</li> <li>REGIME = "regime"</li> <li>EXPERIMENTSTANDARD = "expStandard"</li> <li>SHAPE3D = "shape3D"</li> <li>FORCEPULSE = "forcePulse"</li> </ul> <p>note : déterminer le type de message qui peut être transféré par ce port</p>	<p>1. comme port de communication entre facettes et/ou entre agents</p> <p>2. gérer le fonctionnement de processus proposé par une facette</p>

Classes	Propriétés	Fonctionnement (à propos de communication)
	<ol style="list-style-type: none"> <li>3. <code>setDemon(Function <i>demon</i>)</code> note : définir une fonction qui sera active en suivant l'activation de port</li> <li>4. <code>setDataType(String <i>type</i>)</code> note : définir le type de données transportées par le message à port cible</li> <li>5. <code>setReturnType(String <i>type</i>)</code> note : définir le type de données transportées par le message de port cible</li> <li>6. <code>setMessage(Message <i>msg</i>)</code> note : définir message</li> <li>7. <code>getMessage()</code> note : retourner message</li> <li>8. <code>deleteMessage()</code> note : supprimer message</li> <li>9. <code>active()</code> note : mettre en oeuvre ce port</li> <li>10. <code>getPortType()</code> note : retourner type de port</li> <li>11. <code>getDataType()</code> note : retourner type de données envoyées à port cible</li> <li>12. <code>getReturnType()</code> note : retourner type de données envoyées à port source</li> <li>13. <code>setIsActive(boolean <i>active</i>)</code> note : définir si ce port est prêt à opérer</li> <li>14. <code>getValuelsActive()</code> note : retourner la condition si ce port est prêt à opérer</li> <li>15. <code>setValuelsExport(boolean <i>export</i>)</code> note : définir si ce port est exportable</li> <li>16. <code>getValuelsExport()</code> note : retourner la condition si ce port est exportable</li> <li>17. <code>getAgentName()</code> note : retourner le nom d'agent qui contient ce port</li> <li>18. <code>getFacet()</code> note : retourner facette de ce port</li> </ol>	

Classes	Propriétés	Fonctionnement (à propos de communication)
	<p>19. getTurn() note : retourner la condition de fonctionnement de ce port en cours, comme port source ou comme port cible</p>	
<ul style="list-style-type: none"> <li>PortE note : définir le fonctionnement d'un port cible</li> </ul>	<p>Redéfinir la fonction active() note : supprimer le message en fin d'activation</p>	
<ul style="list-style-type: none"> <li>PortS note : définir le fonctionnement d'un port source</li> </ul>	<p>Redéfinir la fonction active() note : en cours d'activation, envoyer message à facette Contrôle (agent)</p>	
<ul style="list-style-type: none"> <li>PortES note : <ul style="list-style-type: none"> <li>définir le fonctionnement d'un port cible-source</li> <li>première tour, il fonctionne comme un port cible</li> <li>deuxième tour, il fonctionne comme un port source</li> </ul> </li> </ul>	<p>Redéfinir la fonction active()</p>	
<p><i>Function</i> note : interface implémentée par une classe qui définit un processus activé suivant l'activation d'un port</p>	<p>1. functionActive() note : définir un processus</p>	
<p><i>Traduction</i> note : interface implémentée par la classe qui définit la façon dont l'administrateur traite un message</p>	<p>1. translate() note : définir la traduction de message par un administrateur</p>	
<p><i>Message</i></p>	<p>1. setData(Object data) note : définir la donnée (le contenu du message)</p> <p>2. setDestination(String nom_port, String nom_facet, String nom_agent) note : définir la destination du message</p>	<p>1. transporter les données</p> <p>2. définir les destinations des données note : les agents et les facettes où un message passera doivent être définis dans la liste des destinations du message</p>

Classes	Propriétés	Fonctionnement (à propos de communication)
	<ol style="list-style-type: none"> <li>3. <code>setAgentDestination(String nom_agent)</code> note : définir le nom de l'agent destinataire de ce message</li> <li>4. <code>getPortDestinations()</code> note : retourner énumération des ports destination de ce message</li> <li>5. <code>getAgentDestinations()</code> note : retourner énumération des agents destinataires de ce message</li> <li>6. <code>getFacetDestinations()</code> note : retourner énumération des facettes destinataires de ce message</li> <li>7. <code>getData()</code> note : retourner le contenu de ce message</li> <li>8. <code>size()</code> note : retourner nombre des ports destinations</li> <li>9. <code>checkDestinationPort(String nom)</code> note : vérifier si le port défini existe parmi les ports destinations</li> <li>10. <code>checkDestinationFacet(String nom)</code> note : vérifier si la facette définie est disponible parmi les facettes destinations</li> <li>11. <code>checkDestinationAgent(String nom)</code> note : vérifier si l'agent défini est disponible parmi les agents destinations</li> <li>12. <code>deleteDestination(String nom_port, String nom_facet, String nom_agent)</code> note : supprimer les destinations définies</li> <li>13. <code>deleteAgentDestination(String nom)</code> note : supprimer l'agent destination défini</li> <li>14. <code>deleteFacetDestination(String nom)</code> note : supprimer la facette destination définie</li> <li>15. <code>deletePortDestination(String nom)</code> note : supprimer le port destination défini</li> </ol>	

## ANNEXE J : SCENE GRAPH DES COMPOSANTS PRESENTATIONS DES DISPOSITIS ELECTRIQUES

1. *Scene graph* du composant Présentation du moteur à Courant Continu (**PresMotDC**), identique à celui du générateur à Courant Continu (**PresGenDC**)

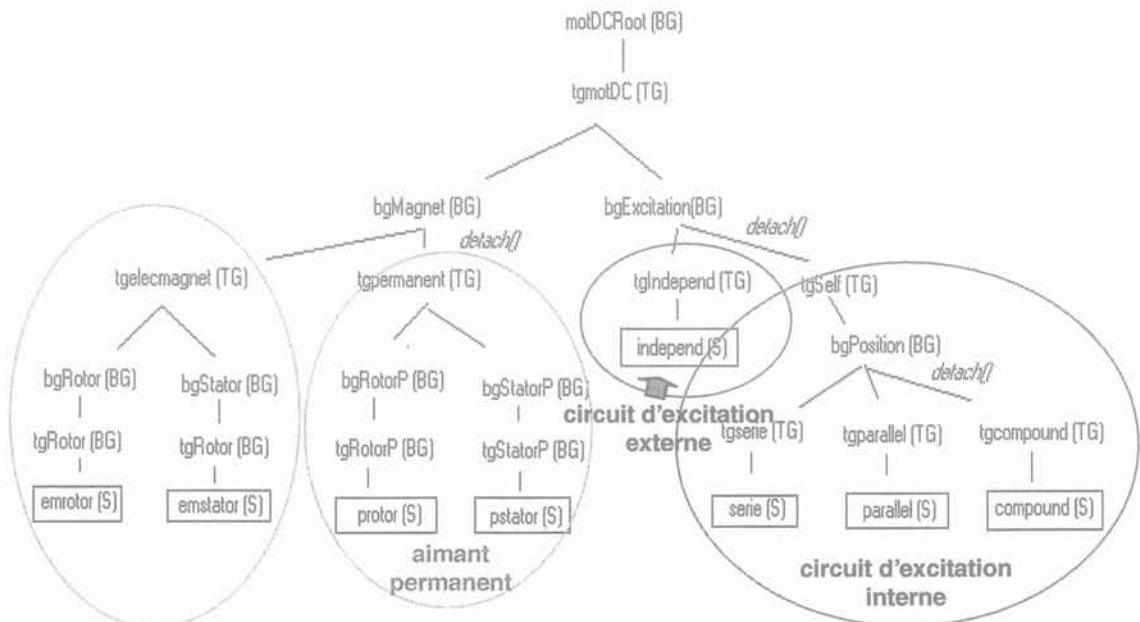


Figure J - 1 *Scene graph* du composant Présentation du moteur à Courant Continu

2. *Scene graph* du composant Présentation de générateur Synchrone (**PresGenSync**), identique à celui du moteur synchrone (**PresMotSync**)

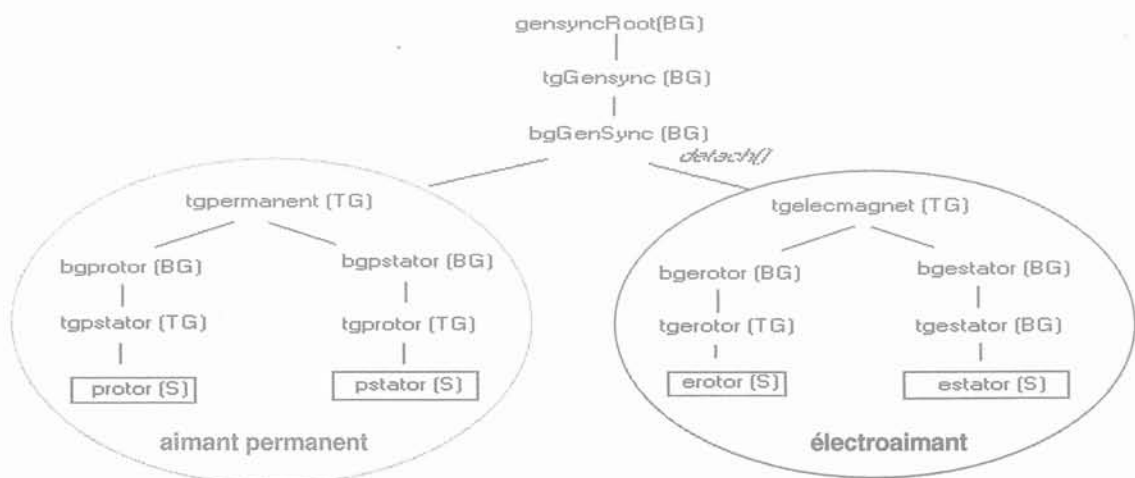


Figure J - 2 *Scene graph* du composant Présentation du générateur Synchrone

3. *Scene graph* du composant Présentation du générateur Asynchrone (**PresGenAsync**), identique à celui du moteur Asynchrone (**PresMotAsync**)

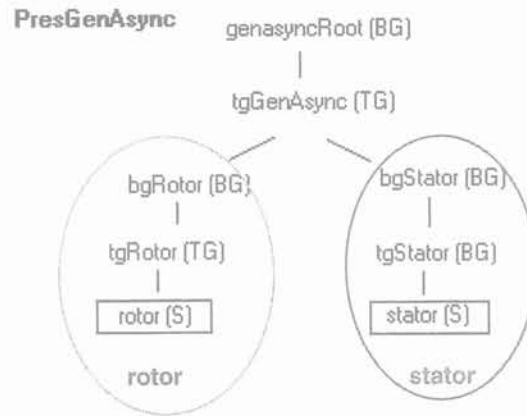


Figure J - 3 *Scene graph* du composant Présentation du générateur Asynchrone

4. *Scene graph* du composant Présentation d'autotransformateur (**PresAutoTrafo**)

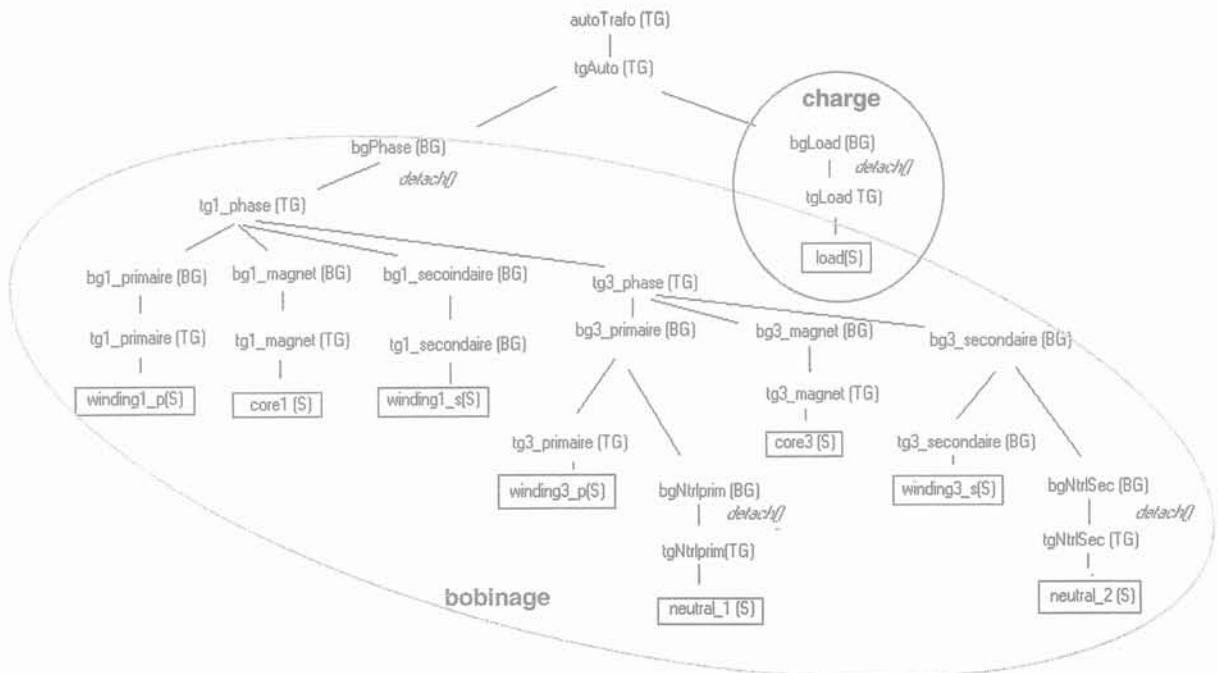


Figure J - 4 *Scene graph* du composant Présentation d'autotransformateur

5. *Scene graph* du composant Présentation du transformateur avec deux bobines (**PresTrafoTwo**), identique à celui du transformateur avec trois bobines (**PresTrafoThree**).

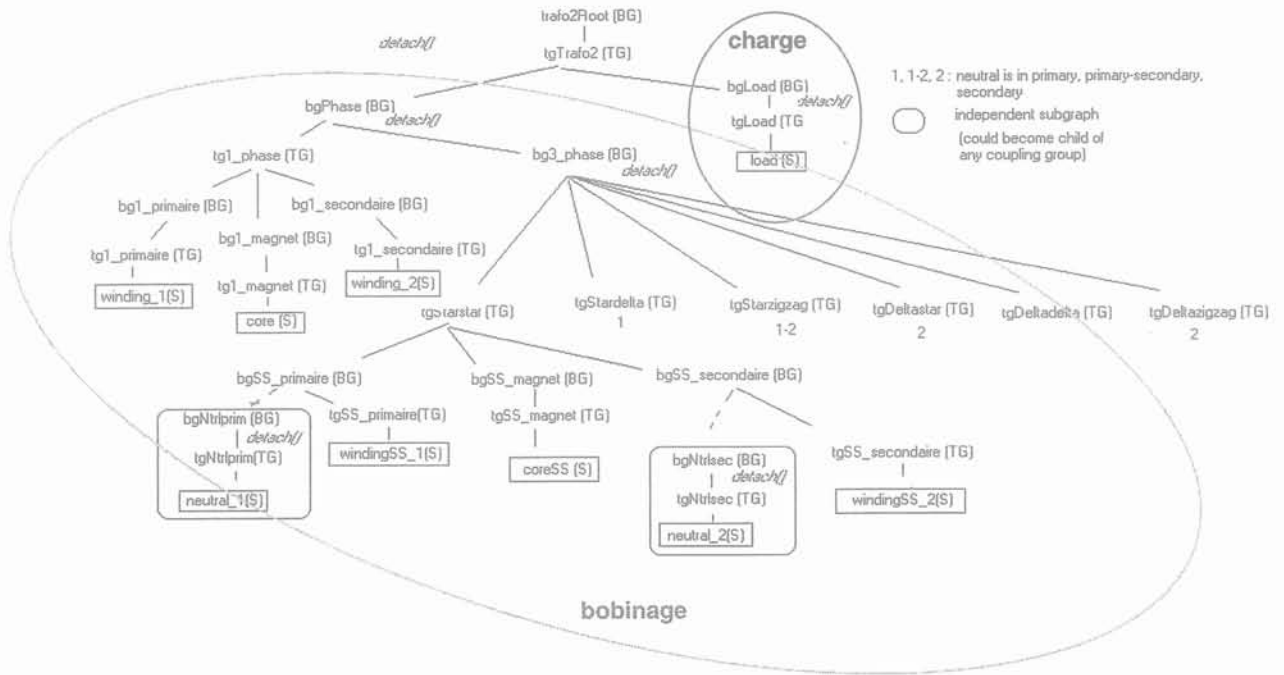


Figure J - 5 *Scene graph* du composant Présentation de transformateur avec deux bobines

## ANNEXE K : ELEMENTS DE ROTERS

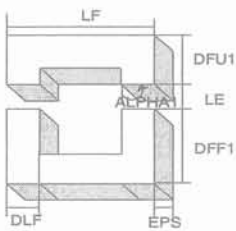
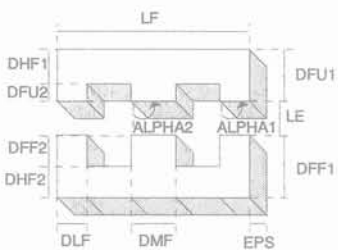
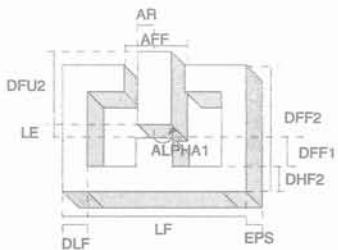
## 1. THEORIE

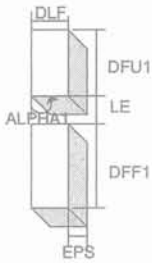
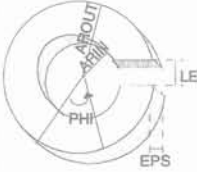
Référence	Explication
Calcul de grandeurs magnétiques en utilisant des coefficients de fuites qui s'appuient sur le calcul de perméances par les éléments de Roters [Roters 41]	<ul style="list-style-type: none"> <li>- Cette méthode considère des flux de fuites aussi bien que le flux utile dans l'entrefer</li> <li>- Le coefficient de fuites de l'électroaimant est égal au total des perméances de l'électroaimant divisé par la perméance du flux utile dans l'entrefer</li> <li>- Le champ magnétique est déterminé en utilisant le théorème de conservation du flux et le théorème d'Ampère</li> </ul> $(1+\nu)B_E S_E = B_F S_F \quad (1)$ $H_E L_E + H_F L_F = NI \quad (2)$ $B_E = \mu_o H_E \quad (3)$ $B_F = \mu_f H_F \quad (4)$ $\frac{B_F S_F L_E}{S_E(1+\nu)} = (NI - H_F L_F) \mu_o \quad (5)$ $H_F = \frac{NI \mu_o}{\left( \frac{\mu_f S_F L_E}{S_E(1+\nu)} + L_F \mu_o \right)} \quad (6)$ <p>où</p> <p><math>\nu</math> : coefficient de fuites de l'électroaimant</p> <p><math>\mu_o</math> : perméabilité de l'entrefer</p> <p><math>\mu_f</math> : perméabilité des tôles</p> <p><math>H_F</math> : champ magnétique dans une tôle</p> <p><math>H_E</math> : champ magnétique dans l'entrefer</p> <p><math>B_F</math> : induction magnétique dans une tôle</p> <p><math>B_E</math> : induction magnétique dans l'entrefer</p> <p><math>S_F</math> : surface d'une tôle</p> <p><math>S_E</math> : surface de l'entrefer</p> <p><math>L_F</math> : ligne moyenne de flux dans une tôle</p> <p><math>L_E</math> : ligne moyenne de flux dans l'entrefer</p> <p><math>NI</math> : ampère tours</p>
Equation de la courbe B(H) pour les matériaux magnétiques non linéaires [Roters 41]	$B(H) = \mu_o H + J_s \frac{H_g + 1 - \sqrt{(1+H_g)^2 - 4H_g(1-\alpha)}}{2(1-\alpha)} \quad (7)$ $H_g = (\mu_{ro} - 1) \frac{\mu_o H}{J_s} \quad (8)$

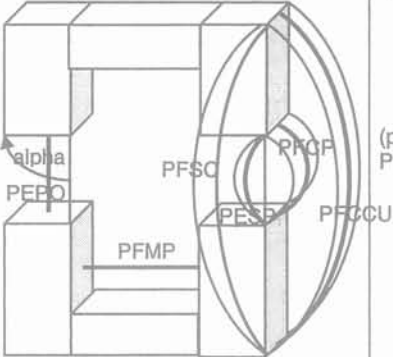


Référence	Explication
	<p>fonction de l'optimisation</p> $F = \sum (B_{\text{Modele}}(H_i) - B_{\text{Mesure}}(i))^2 \quad (9)$ <p>où</p> <p><math>\mu_{ro}</math> : pente à l'origine de la courbe B(H)</p> <p><math>\alpha</math> : coefficient de saturation compris entre 0 et 1</p> <p><math>J_s</math> : l'induction à saturation</p>

## 2. APPLICATION

Entités	Contenu
<p><b>Paramètre constant</b></p> <p>1. GEOSHAPE</p>	<p>MAGNET_CIRCUIT_IN_U = 1</p>  <p>MAGNET_CIRCUIT_IN_E = 2</p>  <p>MAGNET_CIRCUIT_IN_PLUNGER = 3</p> 

Entités	Contenu
	<p>MAGNET_CIRCUIT_IN_I = 4</p>  <p>MAGNET_CIRCUIT_IN_O = 5</p> 
2. PLANSURFACE note : pour circuit magnétique en O	<p>PARALLEL_PLAN_SURFACE = 1 NON_PARALLEL_PLAN_SURFACE = 2</p>
3. magnet_linearity	<p>LINEAR = 1 NON_LINEAR = 2</p>
4. CURRENTYPE	<p>AC_SOURCE = 1 DC_SOURCE = 2</p>
5. PHENOMENA	<p>MAGNETOSTATIC = 1 MAGNETODYNAMIC = 2</p>
6. $\mu_0$ (perméabilité de l'air)	<p><math>4.0 * \text{Math.PI} * 0.0000001</math></p>
<b>Paramètre d'entrée</b>	
1. mur	pente origine de courbe B(H)
2. Js	B saturation
3. LF	largeur de circuit magnétique
4. LE	hauteur de l'entrefer
5. EPS	épaisseur de circuit magnétique
6. DFU	hauteur de pôle mobile - vecteur
7. DFF	hauteur de pôle fixé - vecteur
8. DHF	hauteur de barre horizontale - vecteur

Entités	Contenu
9. DLF	largeur de pôle
10. DMF note : pour circuit magnétique en E	largeur de pôle milieu
11. ALPHA	angle inclination de l'entrefer
12.	
• AFF	distance entre pôles
• AR	rayon de plongeur circulaire
note : pour circuit magnétique avec plongeur	
13.	
• ARIN	rayon interne
• AROUT	rayon externe
• PHI	angle de pôle
note : pour circuit magnétique en O	
<p><b>Fonction</b></p> <p><b>CALCUL DE PERMEANCE</b></p>  <p>Les perméances des différentes lignes moyennes de champ magnétique</p>	<p><b>ligne droite</b></p> <p>PEPO =</p> $UA * width * (((length / \sin(\alpha)) - \text{airgap} * \cos(\alpha)) / (\text{airgap} * \sin(\alpha))) + ((4.0 * \cos(\alpha) * \sin(\alpha)) / ((1.0 + \sin(\alpha)) * (1.0 + \sin(\alpha))))$ <p>(pour armature plongeant)</p> <p>PEPOPLU =</p> $UA * \text{Math.PI} * (((radius / \sin(\alpha)) - \text{airgap} * \cos(\alpha)) / (\text{airgap} * \sin(\alpha))) + ((4.0 * \cos(\alpha) * \sin(\alpha)) / ((1.0 + \sin(\alpha)) * (1.0 + \sin(\alpha)))) * (((radius / \sin(\alpha)) - \text{airgap} * \cos(\alpha)) / (\text{airgap} * \sin(\alpha))) + ((4.0 * \cos(\alpha) * \sin(\alpha)) * \sin(\alpha)) / ((1.0 + \sin(\alpha)) * (1.0 + \sin(\alpha))))$ <hr/> <p><b>un demi cylindre interne</b></p> <p>PFCCP =</p> $0.26 * UA * length / \sin(\alpha)$ <p>(pour armature plongeant)</p> <p>PFCCPLU =</p> $(0.26 * UA * 2.0 * \text{Math.PI} * radius) / \sin(\alpha)$ <hr/> <p><b>un demi cylindre externe</b></p> <p>PFCCU =</p> $UA * 2 * length / (\text{Math.PI} * \sin(\alpha) * ((\text{airgap} / \text{height}) + 1.0))$ <p>(plonge)</p> <p>PFCCUPLU =</p> $(UA * 2.0 * \text{Math.PI} * radius) / (\text{Math.PI} * \sin(\alpha) * ((\text{airgap} / \text{height}) + 1.0))$

Entités	Contenu
	<p><i>un quart de sphère interne</i></p> <p>PFSC =  <math>0.077 * UA * \text{airgap}</math></p> <p><i>un quart de sphère externe</i></p> <p>PFSP =  <math>0.25 * UA * \text{height}</math></p> <hr/> <p><i>entre pôle</i></p> <p>u =  <math display="block">\frac{(\text{distance\_pôle} * \text{distance\_pôle} - \text{radius\_first\_pôle} * \text{radius\_first\_pôle} - \text{radius\_second\_pôle} * \text{radius\_second\_pôle})}{(2.0 * \text{radius\_first\_pôle} * \text{radius\_second\_pôle})}</math></p> <p>PFMP =  <math>2.0 * UA * \text{Math.PI} * \text{width} / (\text{Math.log}(u + \text{Math.sqrt}(u * u - 1)))</math></p> <p>(pour armature plongeant)</p> <p>PFMPLU =  <math display="block">UA * (((\text{Math.PI} * \text{height\_wire\_area}) / 8.0) * (2.0 * \text{radius} + \text{distance\_plu\_branch}) / \text{distance\_plu\_branch} - (2.0 * \text{radius} + \text{distance\_plu\_branch}) + ((2.0 * ((\text{radius} + \text{distance\_plu\_branch}) * (\text{radius} + \text{distance\_plu\_branch}) - (\text{radius} * \text{radius}))) / (\text{Math.PI} * \text{height\_wire\_area})))</math></p> <hr/> <p><b>MAGNET_CIRCUIT_IN_U</b></p> <p><b>Si DFU1 != DHF1</b></p> <p><i>Armature</i></p> <p>p1 = PEPO(EPS, DLF, LE, ALPHA1)  p2 = 2.0 * PFCP(EPS, ALPHA1)  p3 = 2.0 * PFCP(DLF, ALPHA1)  p4 = PFCCU(EPS, 2.0 * ((2.0 * (LF - 2.0 * DLF) / Math.PI) - 0.5 * LE), LE, ALPHA1)  p5 = PFCCU(EPS, 2.0 * ((LF - 2.0 * DLF) / Math.PI) - 0.5 * LE, LE, ALPHA1)  p6 = 2.0 * PFCCU(DLF, 2.0 * ((2.0 * (LF - 2.0 * DLF) / Math.PI) - 0.5 * LE), LE, ALPHA1)  p7 = 4.0 * PFSC(LE)  p8 = 2.0 * PFSP(2.0 * ((2.0 * (LF - 2.0 * DLF) / Math.PI) - 0.5 * LE))  p9 = 2.0 * PFSP(2.0 * ((LF - 2.0 * DLF) / Math.PI) - 0.5 * LE))</p> <p>PA = p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9</p> <p><i>Flux fuite</i></p> <p>PL = 2.0 * PFMP(EPS, LF - 2.0 * DLF, 0.5 * DLF, 0.5 * DLF)</p> <p><i>Flux utile</i></p> <p>PU = p1</p> <p>PTOTAL = PA + PL</p> <p><b>Si DFU1 == DHF1</b></p> <p><i>Armature</i></p> <p>p1 = PEPO(EPS, DLF, LE, 1.57)  p2 = PFCP(EPS, 1.57)  p3 = 2.0 * PFCP(DLF, 1.57)</p>

Entités	Contenu
	<p> <math>p4 = PFCCU(EPS, DFU1 - DHF1, LE, 1.57)</math>  <math>p5 = 2.0 * PFCCU(DLF, DFU1 - DHF1, LE, 1.57)</math>  <math>p6 = 6.0 * PFSC(LE)</math>  <math>p7 = 6.0 * PFSP(DFU - DHF1)</math>  <math>p8 = 2.0 * PFSP(EPS, 1.57)</math>  <math>p9 = 2.0 * PFCCU(EPS, DFU1 - DHF1, LE, 1.57)</math> </p> <p> <math>PA = p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9</math> </p> <p><b>Flux Fuite</b></p> <p> <math>PL = 2.0 * PFMP(EPS, LF - 2.0 * DLF, 0.5 * DLF, 0.5 * DLF)</math> </p> <p><b>Flux Utile</b></p> <p> <math>PU = p1</math> </p> <p> <math>PTOTAL = PA + PL</math> </p> <p><b>MAGNET_CIRCUIT_IN_I</b></p> <p><b>Armature</b></p> <p> <math>p1 = PEPO(EPS, DLF, LE, ALPHA1)</math>  <math>p2 = 2.0 * PFSP(EPS, ALPHA1)</math>  <math>p3 = 2.0 * PFSP(DLF, ALPHA1)</math>  <math>p4 = 2.0 * PFCCU(EPS, DFU1, LE, ALPHA1)</math>  <math>p5 = 2.0 * PFCCU(DLF, DFU1, LE, ALPHA1)</math>  <math>p6 = 4.0 * PFDC(LE)</math>  <math>p7 = 4.0 * PFSP(DFU1)</math> </p> <p> <math>PA = p1 + p2 + p3 + p4 + p5 + p6 + p7</math> </p> <p><b>Flux fuite</b></p> <p> <math>PL = 0.0</math> </p> <p><b>Flux utile</b></p> <p> <math>PU = p1</math> </p> <p> <math>PTOTAL = PA + PL</math> </p> <hr/> <p><b>MAGNET_CIRCUIT_IN_E</b></p> <p><b>SI DFU1 != DHF1</b></p> <p><b>Armature</b></p> <p><b>Externe</b></p> <p> <math>p1 = PEPO(EPS, DLF, LE, ALPHA1)</math>  <math>p2 = 2.0 * PFSP(EPS, ALPHA1)</math>  <math>p3 = 2.0 * PFSP(DLF, ALPHA1)</math>  <math>p4 = PFCCU(EPS, 2.0 * ((2.0 * 0.5 * (LF - 2.0 * DLF - DMF) / Math.PI) - 0.5 * LE), LE, ALPHA1)</math>  <math>p5 = PFCCU(EPS, 2.0 * ((0.5 * (LF - 2.0 * DLF - DMF) / Math.PI) - 0.5 * LE), LE, ALPHA1)</math>  <math>p6 = 2.0 * PFCCU(DLF, 2.0 * ((2.0 * 0.5 * (LF - 2.0 * DLF - DMF) / Math.PI) - 0.5 * LE), LE, ALPHA1)</math>  <math>p7 = 4.0 * PFSC(LE)</math>  <math>p8 = 2.0 * PFSP(2.0 * ((2.0 * 0.5 * (LF - 2.0 * DLF - DMF) / Math.PI) - 0.5 * LE))</math>  <math>p9 = 2.0 * PFSP(2.0 * ((0.5 * (LF - 2.0 * DLF - DMF) / Math.PI) - 0.5 * LE))</math> </p> <p> <math>Po = p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9</math> </p>

Entités	Contenu
	<p><b>Flux utile</b></p> <p><math>PU = p1</math></p> <p><b>Armature</b></p> <p><i>Interne</i></p> <p> <math>p1 = PEPO(EPS, 0.5 \cdot DMF, LE, ALPHA2)</math>  <math>p2 = PFCP(EPS, ALPHA2)</math>  <math>p3 = 2.0 \cdot PFCP(0.5 \cdot DMF, ALPHA2)</math>  <math>p4 = 2.0 \cdot PFCCU(0.5 \cdot DMF, 2.0 \cdot ((2.0 \cdot (0.5 \cdot LF - (DLF + 0.5 \cdot DMF))) / \text{Math.PI} - (0.5 \cdot LE))), LE, ALPHA2)</math>  <math>p5 = PFCCU(EPS, 2.0 \cdot ((2.0 \cdot (0.5 \cdot LF - (DLF + 0.5 \cdot DMF))) / \text{Math.PI} - (0.5 \cdot LE))), LE, ALPHA2)</math>  <math>p6 = 2.0 \cdot PFSC(LE)</math>  <math>p7 = 2.0 \cdot PFSP((0.5 \cdot LF - (DLF + 0.5 \cdot DMF)) / \text{Math.PI} - (0.5 \cdot LE))</math> </p> <p><math>Pi = p1 + p2 + p3 + p4 + p5 + p6 + p7</math></p> <p><math>PA = (Po \cdot Pi) / (Po + Pi)</math></p> <p><b>Flux fuite</b></p> <p><math>PL = PFMP(EPS, 0.5 \cdot LF - (DLF + 0.5 \cdot DMF), 0.5 \cdot DLF, 0.5 \cdot DMF)</math></p> <p><b>Si DFU1 == DHF1</b></p> <p><b>Armature</b></p> <p><i>Externe</i></p> <p> <math>p1 = PEPO(EPS, DLF, LE, 1.57)</math>  <math>p2 = PFCP(EPS, 1.57)</math>  <math>p3 = 2.0 \cdot PFCP(DLF, 1.57)</math>  <math>p4 = PFCCU(EPS, DFU1 - DHF1, LE, 1.57)</math>  <math>p5 = 2.0 \cdot PFCCU(DLF, DFU1 - DHF1, LE, 1.57)</math>  <math>p6 = 6.0 \cdot PFSC(LE)</math> </p> <p> <math>p7 = 6.0 \cdot PFSP(DFU1 - DHF1)</math>  <math>p8 = 2.0 \cdot PFCP(EPS, 1.57)</math>  <math>p9 = 2.0 \cdot PFCCU(EPS, DFU1 - DHF1, LE, 1.57)</math> </p> <p><math>Po = p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9</math></p> <p><b>Flux utile</b></p> <p><math>PU = p1</math></p> <p><b>Armature</b></p> <p><i>Interne</i></p> <p> <math>p1 = PEPO(EPS, DMF, LE, 1.57)</math>  <math>p2 = 2.0 \cdot PFCP(0.5 \cdot DMF, 1.57)</math>  <math>p3 = 2.0 \cdot PFCCU(0.5 \cdot DMF, DFU2 - DHF1, LE, 1.57)</math>  <math>p4 = PFCP(EPS, 1.57)</math>  <math>p5 = PFCCU(EPS, DFU2 - DHF1, LE, 1.57)</math>  <math>p6 = 4.0 \cdot PFSC(LE)</math>  <math>p7 = 5.0 \cdot PFSP(DFU2 - DHF1)</math> </p> <p><math>Pi = p1 + p2 + p3 + p4 + p5 + p6 + p7</math></p> <p><math>PA = (Pi \cdot Po) / (Pi + Po)</math></p>

Entités	Contenu
	<p><b>Flux fuite</b></p> $PL = PFMP(EPS, LF - (DLF + 0.5 \cdot DMF), 0.5 \cdot DLF, 0.5 \cdot DMF)$ $PTOTAL = PA + PL$ <p><b>MAGNET_CIRCUIT_IN_PLUNGER</b></p> <p><b>Armature</b></p> $p1 = PEPOPLU(AR, LE, ALPHA1)$ $p2 = PFCPLU(AR, ALPHA1)$ $p3 = PFCCUPLU(AR, ((2.0 \cdot (0.5 \cdot LF - DLF - AR)) / \text{Math.PI}) - 0.5 \cdot LE, LE, ALPHA1)$ $PA = p1 + p2 + p3$ <p><b>Flux utile</b></p> $PU = p1$ <p><b>Flux fuite</b></p> $PL = PFMPLU(AR, DFF1 - DHF2, 0.5 \cdot LF - AR - DLF)$ $PTOTAL = PA + PL$
CALCUL DE COEFFICIENT DE FUITE	<p><b>Armature</b></p> $LAMDA\_ARM = PA/PU$ <p><b>Flux fuite</b></p> $LAMDA\_LEA = PL/PU$ <p><b>Circuit Magnétique</b></p> $LAMDA\_MAG = PTOTAL/PU$
CALCUL D'AMPERE TOUR	$NI = N \cdot (\text{power/voltage})$
CALCUL DE LIGNE MOYENNE  note : SA :surface de fonctionnement de l'entrefer LA :ligne moyenne de l'entrefer SM :surface de circuit magnétique LM :ligne moyenne de circuit magnétique	<p><b>MAGNET_CIRCUIT_IN_E</b></p> <p><b>Armature</b></p> <p><b>Si DFU1 != DHF1</b></p> $LMU = 0.5 \cdot LF - 0.5 \cdot DLF - 0.25 \cdot DMF + DFU1 - 0.5 \cdot DHF1 + DFU2 + 0.5 \cdot DHF1$ <p><b>Si DFU1 == DHF1</b></p> $LMU = 0.5 \cdot LF - 0.5 \cdot DLF - 0.25 \cdot DMF$ <p><b>Fixé</b></p> <p><b>Si DFF1 != DHF2</b></p> $LMF = 0.5 \cdot LF - 0.5 \cdot DLF - 0.25 \cdot DMF + DFF1 - 0.5 \cdot DHF2 + DFF2 + 0.5 \cdot DHF2$

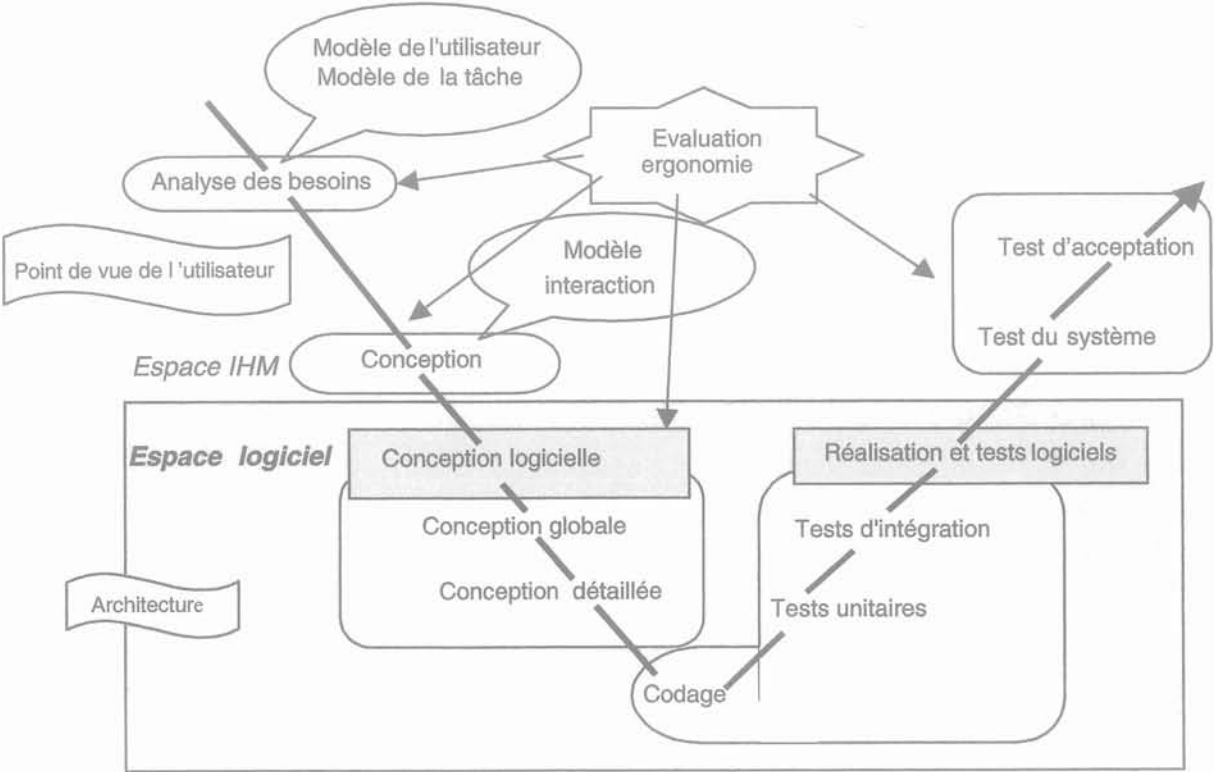
Entités	Contenu
	<p><b>SI DFF1 == DHF2</b></p> <p> <math>LMF = 0.5 * LF - 0.5 * DLF - 0.25 * DMF</math>  <math>LM = LMU + LMF</math>  <math>SM = 0.5 * EPS * (DLF + 0.5 * DMF)</math>  <math>SA = EPS * DLF</math>  <math>LA = 2.0 * LE</math> </p> <hr/> <p><b>MAGNET_CIRCUIT_IN_I</b></p> <p> <math>LM = DFU + DFF</math>  <math>SM = EPS * DLF</math>  <math>SA = SM</math>  <math>LA = LE</math> </p> <hr/> <p><b>MAGNET_CIRCUIT_IN_PLUNGER</b></p> <p><i>Armature</i></p> <p><math>LMU = DFU</math></p> <p><i>Fixé</i></p> <p> <math>LMF = DFF1 - DHF2 + 0.5 * LF - AR - 0.5 * DLF + DFF1 - DHF2 + 0.5 * DHF1 + 0.5 * LF - 0.5 * DLF - 0.5 * AFF</math> </p> <p> <math>LM = LMU + LMF</math>  <math>SM = Math.PI * AR * AR</math>  <math>SA = SM</math>  <math>LA = LE</math> </p> <hr/> <p><b>MAGNET_CIRCUIT_IN_U</b></p> <p><i>Armature</i></p> <p><b>SI DFU1 != DHF1</b></p> <p><math>LMU = 0.5 * LF - DLF + DFU1 - 0.5 * DHF1 + DFU1 - 0.5 * DHF1</math></p> <p><b>SI DFU1 == DHF1</b></p> <p><math>LMU = 0.5 * LF - DLF</math></p> <p><i>Fixé</i></p> <p><b>SI DFF1 != DHF2</b></p> <p><math>LMF = 0.5 * LF - DLF + DFF1 - 0.5 * DHF2 + DFF1 - 0.5 * DHF2</math></p> <p><b>SI DFF1 == DHF2</b></p> <p> <math>LMF = 0.5 * LF - DLF</math>  <math>LM = LMU + LMF</math> </p> <p> <math>SM = EPS * DLF</math>  <math>SA = SM</math>  <math>LA = 2.0 * LE</math> </p>



Entités	Contenu
<p><b>CREATION DE COURBE B(H)</b></p> <p>(CALCUL COEFFICIENT DE SATURATION)</p> <pre> graph TD     Input[/mu_o J_s n, fa points B(H) alpha(a,b), err/] --&gt; Hg[Calcul de Hg]     Hg --&gt; Bm[Calcul de Bm]     Bm --&gt; F[Calcul de F]     F --&gt; Decision{F &lt;= err ou i &gt; n}     Decision -- oui --&gt; Alpha[alpha]     Decision -- no --&gt; Add[alpha + fa]     Add --&gt; Bm   </pre> <p><b>NEWTON RAPHSON</b></p> <pre> graph TD     Input[/Ho err itération/] --&gt; Calc[Calcul H = Ho * F(Ho) / F'(Ho)]     Calc --&gt; Decision{(H - Ho) / H &lt;= err ou i &gt; itération}     Decision -- oui --&gt; HF[HF = H]     Decision -- no --&gt; Inc[i++]     Inc --&gt; HoH[Ho = H]     HoH --&gt; Calc   </pre>	<p> <math>s = 0.001</math>      valeur initiale de a (coefficient de saturation)  <math>fa = 0.0001</math>    facteur additionnel de a  <math>F = 0</math>            correction d'erreur   </p> <pre> while(!stop){   Hg = (mur - 1.0)*(mo*pointH.elementAt(i)/js)   Bm = mo*pointH.elementAt(i) + js*((Hg + 1.0 - Math.sqrt((1.0+Hg)*(1.0+Hg)-4.0*Hg*(1.0 - s)))/(2.0*(1-s)))   F = F + (Bm - pointB.elementAt(i))*(Bm - pointB.elementAt(i));   i++;    if(F &lt;= err    i &gt; n){     stop = true     alpha = s   }   s = s + fa }   </pre> <hr/> <p> <i>fonction</i> CALCUL D'AMPERE TOUR  <i>fonction</i> CALCUL DE LIGNE MOYENNE   </p> <p> <math>Ho = 0.0</math>  while(!convergence){ <math display="block">F = mo*Ho + (js*((mo*mur*Ho/js) - (mo*Ho/js) + 1.0 - Math.sqrt((1.0 + (mo*mur*Ho/js) - (mo*Ho/js))*(1.0 + (mo*mur*Ho/js) - (mo*Ho/js)) - (4.0*mo*Ho/js)*(mur - 1.0 - mur*a + a)))/(2.0*(1.0 - a))) - ((NI - Ho*LM)*UA*SA*(1 + LAMDA\_MAG)/(SM*LA))</math> <math display="block">F1 = mo + (js*((mo*mur/js) - (mo/js) - (0.5/(Math.sqrt(1 + (mo*mur*Ho/js) - (mo*Ho/js))*(1.0 + (mo*mur*Ho/js) - (mo*Ho/js)) - (4.0*mo*Ho/js)*(mur - 1.0 - mur*a + a))))*(2.0*(1.0 + (mo*mur*Ho/js) - (mo*Ho/js)))/((mo*mur/js) - (mo/js)) - (4.0*mo/js)*(mur - 1.0 - mur*a + a)))/(2.0*(1.0 - a))) + (UA*SA*(1 + LAMDA\_MAG)*LM/(SM*LA))</math> <math>H = Ho - (F/F1)</math> if(Math.abs((H - Ho)/H) &lt;= err    i == iteration){ convergence = true HF = H } i++ Ho = H } <math display="block">BF = ((NI - HF*LM)*UA*SA*(1 + LAMDA\_MAG))/(SM*LA))</math> </p>

Entités	Contenu
<b>CALCUL B_H D'AIMANT LINEAIRE</b>	<i>fonction</i> CALCUL D'AMPERE TOUR <i>fonction</i> CALCUL DE LIGNE MOYENNE $HF = (NI \cdot UA) / (((UM \cdot SM \cdot LA) / LAMDA\_MAG) + LM \cdot UA)$ $BF = UM \cdot HF$ <hr/>
<b>CALCUL B_H D'AIMANT NON LINEAIRE</b>	<i>fonction</i> CREATION DE COURBE B(H) <i>fonction</i> NEWTON RAPHSON

# ANNEXE L : METHODE DE CONCEPTION ET CYCLE EN V



## ANNEXE M : LES NOMS DES DONNEES CREEES AU COURS DE L'EXPERIMENTATION (DE CONTACTEUR) DANS LVEE

Agent	Gestion des données	Données		Paramètres dans facette Présentation
		Nom	Valeur	
apparaillage	classe Parameter	DEVICSource	AC DC AC/DC	ContactorCharacterDialog.source
		CONTACTORACVoltSupply		unac
		CONTACTORDCVoltSupply		undc
		CONTACTORACPowerSupply(open)		acsa
		CONTACTORDCPowerSupply(open)		dcsa
		CONTACTORACPowerSupply(close)		acsm
		CONTACTORDCPowerSupply(close)		dcsm
		CONTACTORCurrentNominal		in
		CONTACTORCurrentThermal		lth
		CONTACTORPoleNumber		pole
		CONTACTORAuxPoleNumber		polesux
		CONTACTORApplication		emploi
		CONTACTORManuver		manuver
		CONTACTORStiffnessMagnetSpring		RappelSpringDialog.stiffness
		CONTACTORMaterialMagnetSpring		material
		CONTACTORStiffnessPoleSpring		PoleSpringDialog.stiffness
		CONTACTORMaterialPoleSpring		material
cont_relai_E		CONTACTORMagnetLayer		PresContE.EM_layer
		CONTACTORHeightMobileBranch		PresContE.Em_Ahm
		CONTACTORHeightMobileHorBar		PresContE.EM_Acm
		CONTACTORLengthSideBranch		PresContE.EM_Acf
		CONTACTORDistanceBetweenBranches		PresContE.EM_Aff
		CONTACTORLengthMiddleBranch		PresContE.EM_Adf
		CONTACTORAirgapDistance		PresContE.EM_Aen
		CONTACTORAirgapMinimum		PresContE.EM_Cm
		CONTACTORHeightFixedBranch		PresContE.EM_Ahl
		CONTACTORHeightFixedHorBar		PresContE.EM_Adm
		CONTACTORLengthMagnet		PresContE.EM_Alf
		CONTACTORWidthMagnet		PresContE.EM_Eps
		CONTACTORMaterialEM		PresContE.EM_material
		CONTACTORLinearityBH		PresContE.EM_LinearityBH
		CONTACTORMaterialAG		PresContE.AG_material
		CONTACTORStrokeAmature		PresContE.EM_C
		CONTACTORShadeCoilDistance		PresContE.SC_Fg
		CONTACTORLengthShadeCoil		PresContE.SC_Fgl
		CONTACTORHeightShadeCoil		PresContE.SC_Fgh

Agent	Gestion des données	Données		Parametres dans facette Présentation
		Nom	Valeur	
		CONTACTORRadiusBorderShadeCoil		PresContE.SC_Rg
		CONTACTORSCLengthCoil		PresContE.SC_FgL
		CONTACTORSCTurn		PresContE.SC_FgN
		CONTACTORSCDiameter		PresContE.SC_FgD
		CONTACTORMaterialSC		PresContE.SC_material
		CONTACTORHorDistanceWire		PresContE.W_Asb
		CONTACTORVerDistanceWire		PresContE.W_Abb
		CONTACTORLengthWire		PresContE.W_Alb
		CONTACTORHeightWire		PresContE.W_Ahb
		CONTACTORWTurn		PresContE.W_N
		CONTACTORWLengthCoil		PresContE.W_L
		CONTACTORWDiameter		PresContE.W_D
		CONTACTORMaterialW		PresContE.W_material
		CONTACTORWireType		PresContE.W_type
		CONTACTOREcrasementContact		PresContE.C_Co
		CONTACTORRadiusContact		PresContE.C_Rc
		CONTACTORThicknessContact		PresContE.C_Ec
		CONTACTORMaterialC		PresContE.C_material
cont_relai_U		CONTACTORHeightMobileBranch		PresContU.EM_Ahm
		CONTACTORHeightMobileHorBar		PresContU.EM_Acm
		CONTACTORLengthSideBranch		PresContU.EM_Acl
		CONTACTORDistanceBetweenBranches		PresContU.EM_Alf
		CONTACTORAirgapDistance		PresContU.EM_Aen
		CONTACTORHeightFixedBranch		PresContU.EM_Ahf
		CONTACTORHeightFixedHorBar		PresContU.EM_Adm
		CONTACTORLengthMagnet		PresContU.EM_All
		CONTACTORWidthMagnet		PresContU.EM_Eps
		CONTACTORMaterialEM		PresContU.EM_material
		CONTACTORLinearityBH		PresContU.EM_linearityBH
		CONTACTORMagnetLayer		PresContU.EM_layer
		CONTACTORMaterialAG		PresContU.AG_material
		CONTACTORStrokeAmature		PresContU.EM_C
		CONTACTORAirgapMinimum		PresContU.EM_Cm
		CONTACTORShadeCoilDistance		PresContU.SC_Fg
		CONTACTORLengthShadeCoil		PresContU.SC_Fgl
		CONTACTORHeightShadeCoil		PresContU.SC_Fgh
		CONTACTORRadiusBorderShadeCoil		PresContU.SC_Rg
		CONTACTORSCLengthCoil		PresContU.SC_FgL
		CONTACTORSCTurn		PresContU.SC_FgN
		CONTACTORSCDiameter		PresContU.SC_FgD

Agent	Gestion des donnés	Données		Paramètres dans facette Présentation
		Nom	Valeur	
		CONTACTORMaterialSC		PresContU.SC_material
		CONTACTORHorDistanceWire		PresContU.W_Asb
		CONTACTORVerDistanceWire		PresContU.W_Abb
		CONTACTORLengthWire		PresContU.W_Alb
		CONTACTORHeightWire		PresContU.W_Ahb
		CONTACTORWTurn		PresContU.W_N
		CONTACTORWLengthCoil		PresContU.W_L
		CONTACTORWDiameter		PresContU.W_D
		CONTACTORMaterialW		PresContU.W_material
		CONTACTORWireType		PresContU.W_type
		CONTACTOREcrasementContact		PresContU.C_Co
		CONTACTORRadiusContact		PresContU.C_Rc
		CONTACTORThicknessContact		PresContU.C_Ec
		CONTACTORMaterialC		PresContU.C_material
cont_relai_P		CONTACTORHeightPlunger		PresContP.EM_Ahm
		CONTACTORLengthPlunger		PresContP.EM_Acm
		CONTACTORHeightMiddleBranch		PresContP.EM_Adif
		CONTACTORLengthSideBranch		PresContP.EM_Acf
		CONTACTORDistanceBetweenHorBranches		PresContP.EM_Alf
		CONTACTORAirgapDistance		PresContP.EM_Aen
		CONTACTORHeightSideBranch		PresContP.EM_Ahlp
		CONTACTORHeightFixedHorBar		PresContP.EM_Adm
		CONTACTORLengthUpperBar		PresContP.EM_Alm
		CONTACTORHeightMagnet		PresContP.EM_Ahi
		CONTACTORLengthMagnet		PresContP.EM_Ail
		CONTACTORWidthMagnet		PresContP.EM_Eps
		CONTACTORMaterialEM		PresContP.EM_material
		CONTACTORLinearityBH		PresContP.EM_linearityBH
		CONTACTORMagnetLayer		PresContP.EM_layer
		CONTACTORMaterialAG		PresContP.AG_material
		CONTACTORStokeAmature		PresContP.EM_C
		CONTACTORAirgapMinimum		PresContP.EM_Cm
		CONTACTORShadeCoilDistance		PresContP.SC_Fg
		CONTACTORLengthShadeCoil		PresContP.SC_Fgl
		CONTACTORHeightShadeCoil		PresContP.SC_Fgh
		CONTACTORRadiusBorderShadeCoil		PresContP.SC_Rg
		CONTACTORSCLengthCoil		PresContP.SC_FgL
		CONTACTORSCTurn		PresContP.SC_FgN
		CONTACTORSCDiameter		PresContP.SC_FgD

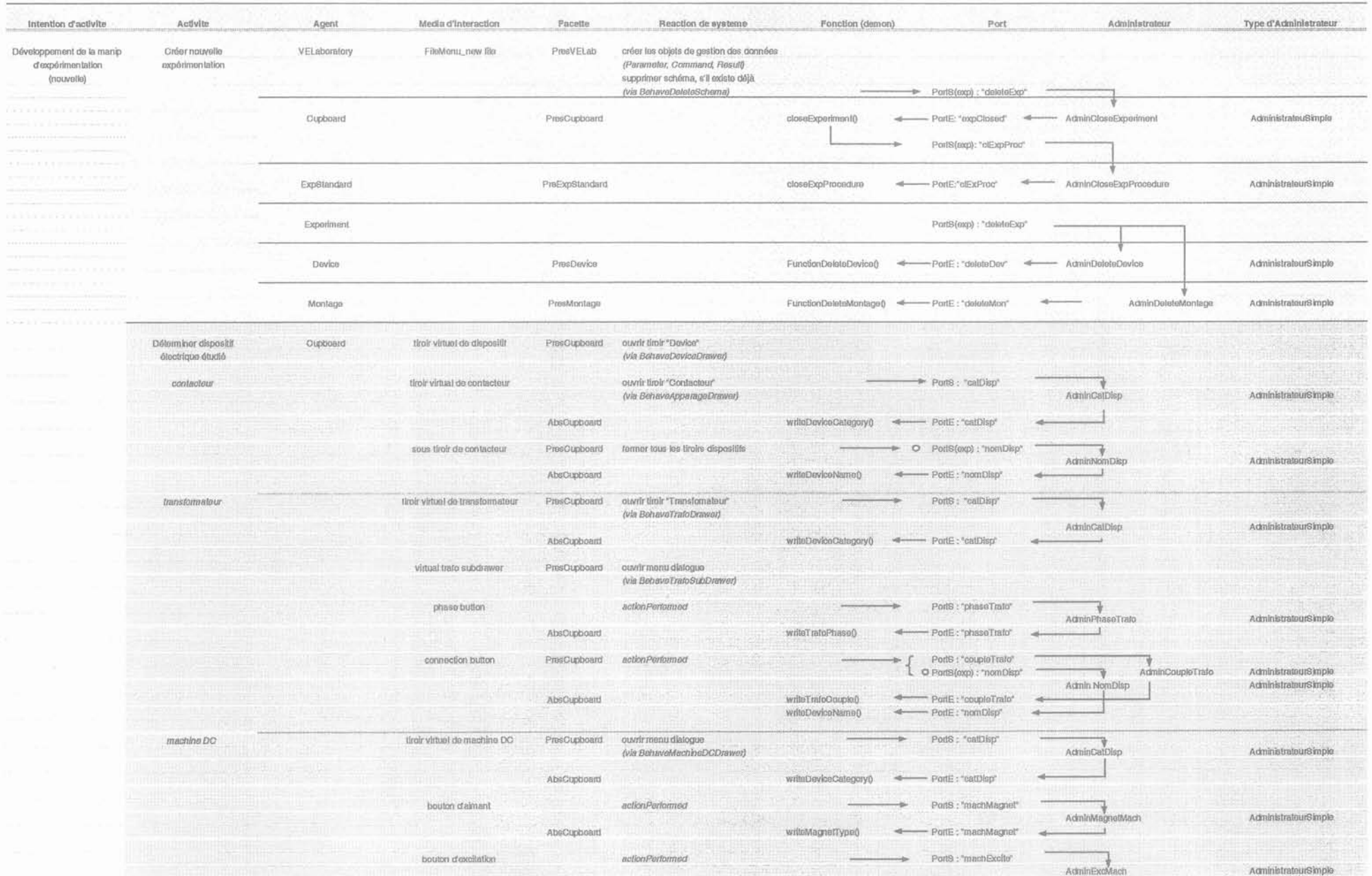
Agent	Gestion des données	Données		Parametres dans facette Présentation
		Nom	Valeur	
		CONTACTORMaterialSC		PresContP.SC_material
		CONTACTORHorDistanceWire		PresContP.W_Asb
		CONTACTORVerDistanceWire		PresContP.W_Abb
		CONTACTORLengthWire		PresContP.W_Alb
		CONTACTORHeightWire		PresContP.W_Ahb
		CONTACTORWTurn		PresContP.W_N
		CONTACTORWLengthCoil		PresContP.W_L
		CONTACTORWDiameter		PresContP.W_D
		CONTACTORMaterialW		PresContP.W_material
		CONTACTORWireType		PresContP.W_type
		CONTACTOREcrasementContact		PresContP.C_Co
		CONTACTORRadiusContact		PresContP.C_Rc
		CONTACTORThicknessContact		PresContP.C_Ec
		CONTACTORMaterialC		PresContP.C_material
cupboard		categorie_dispositif	appareillage transformateur machine_sync machine_async machine_DC	
		nom_dispositif	cont_relai_E cont_relai_P cont_relai_U  trafo_2_enroulement autotrafo trafo_3_enroulement  motor_Async generator_Async motor_Async_pas_a_pas  motor_Sync generator_Sync motor_Sync_pas_a_pas  motor_DC generator_DC motor_DC_pas_a_pas	
		categorie_objectif	trafo_objectif machine_objectif cont_relai_objectif	
		nom_objectif	force_residu force_rappel force_masse force_courant temps_reponse coupure endurance_mecanique propiete_tole chaud_perma chaud_court  frequence_industrie tension_induite chocs_electrique decharge chaud_perma chaud_court eittancheite	

Agent	Gestion des données	Données		Parametres dans facette Présentation
		Nom	Valeur	
			frequence_industrie tension_induite chaud_perma chaud_court chaud_surintensite rendement_magnet perte_mecanique couple_vitesse chocs_mecanique degre_protection intensite_vibratoire niveau_bruit couple_demarrage couple_dechrochage couple_residu essai_glissement couple_maintien facteur_puissance courant_fourni_absorbe rendement resistance_inductance variation_tension harmonique courbe_excitation diverse_reactance courant_court_circuit chute_tension	
		regime	permanent transitoire	
		charge	encharge vide court	
		mach_aimant	permanent nonpermanent	
		mach_excitation	interne independant	
		mach_circuit_excitation	serie paralel compound	
		trafoPhase		1 3
		trafoCouplage	etoile-etoile etoile-zigzag triangle-etoile triangle-zigzag etoile-triangle triangle-triangle  etoile-etoile-etoile etoil-etoile-zigzag triangle-triangle-etoile triangle-triangle-zigzag etoile-etoile-triangle triangle-triangle-triangle	
montage		SOURCECurrentAC		
electricalsource		SOURCECurrentDC		
		SOURCEVoltageDC		
		SOURCEVoltageAC		
		SOURCEFrequency		
		SOURCEPowerFactor		
		SOURCEWaveType	sinusoidal triangle square	



Agent	Gestion des données	Données		Paramètres dans facette Présentation
		Nom	Valeur	
<i>method</i>		SOURCENature	DC (2) AC(1)	
		METHODerreur		
		METHODiteration		
		METHODname	Elements_Finis_2D Elements_Finis_3D Elements_Rotors	
		METHODmaillage		
<i>resultat de simulation</i>	classe Result	MAGNETIC_FIELD	H	
		MAGNETIC_FLUX_DENSITY	B	
VELaboratory <i>pilot</i>	classe Command	EXPCREATEDEV	create_device	
		VELCREATECIR	create_schema	
		EXPCREATEMON	create_montage	
		CONTPODELETE	delete_poles	
		CONTSCDELETE	delete_shade_coil	
		CONTCODELETE	delete_contactor_frame	

# ANNEXE N : SCHEMA DU MECANISME DE COMMUNICATION ENTRE LES AGENTS INTERACTIFS DU LVEE

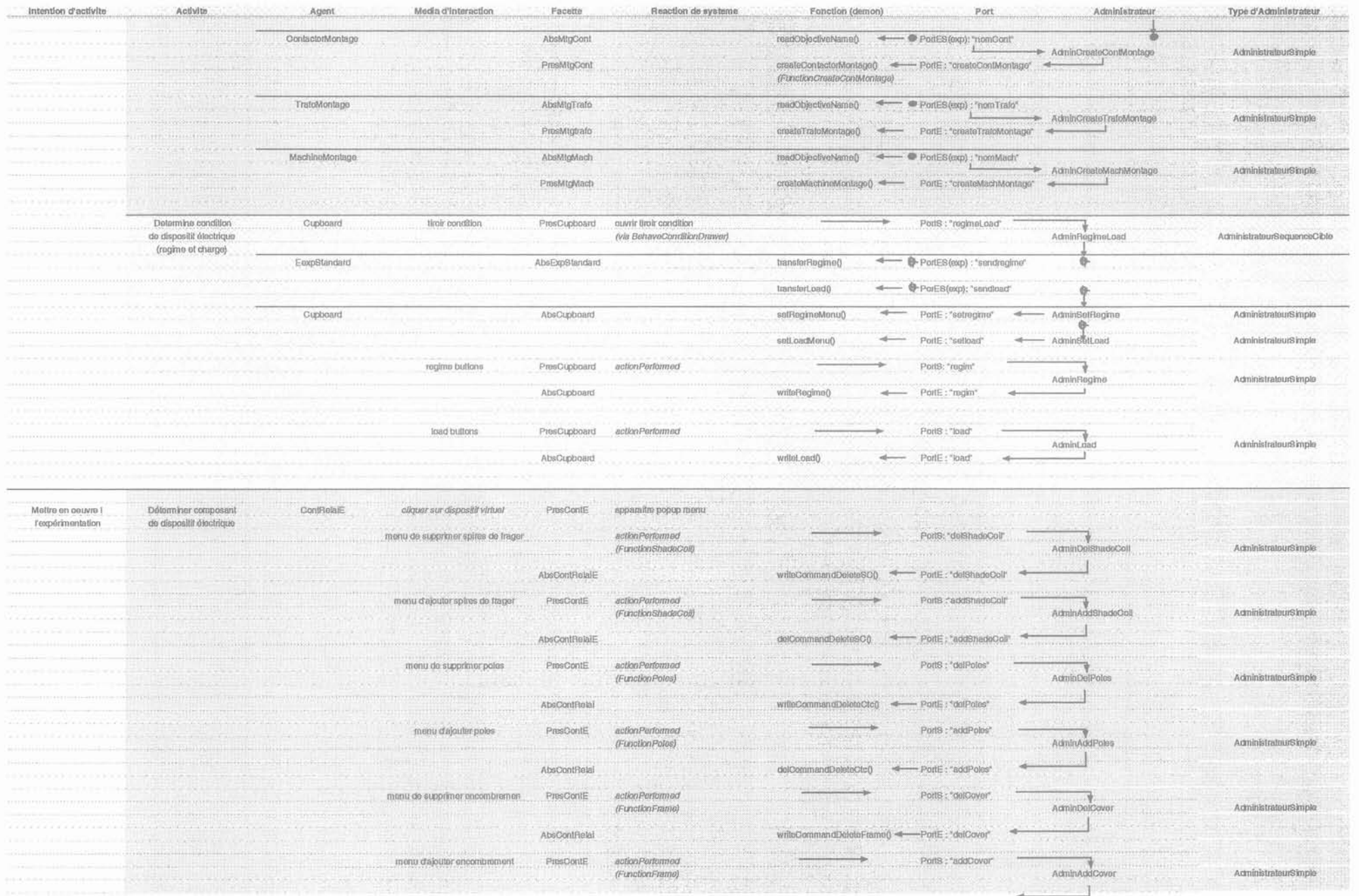


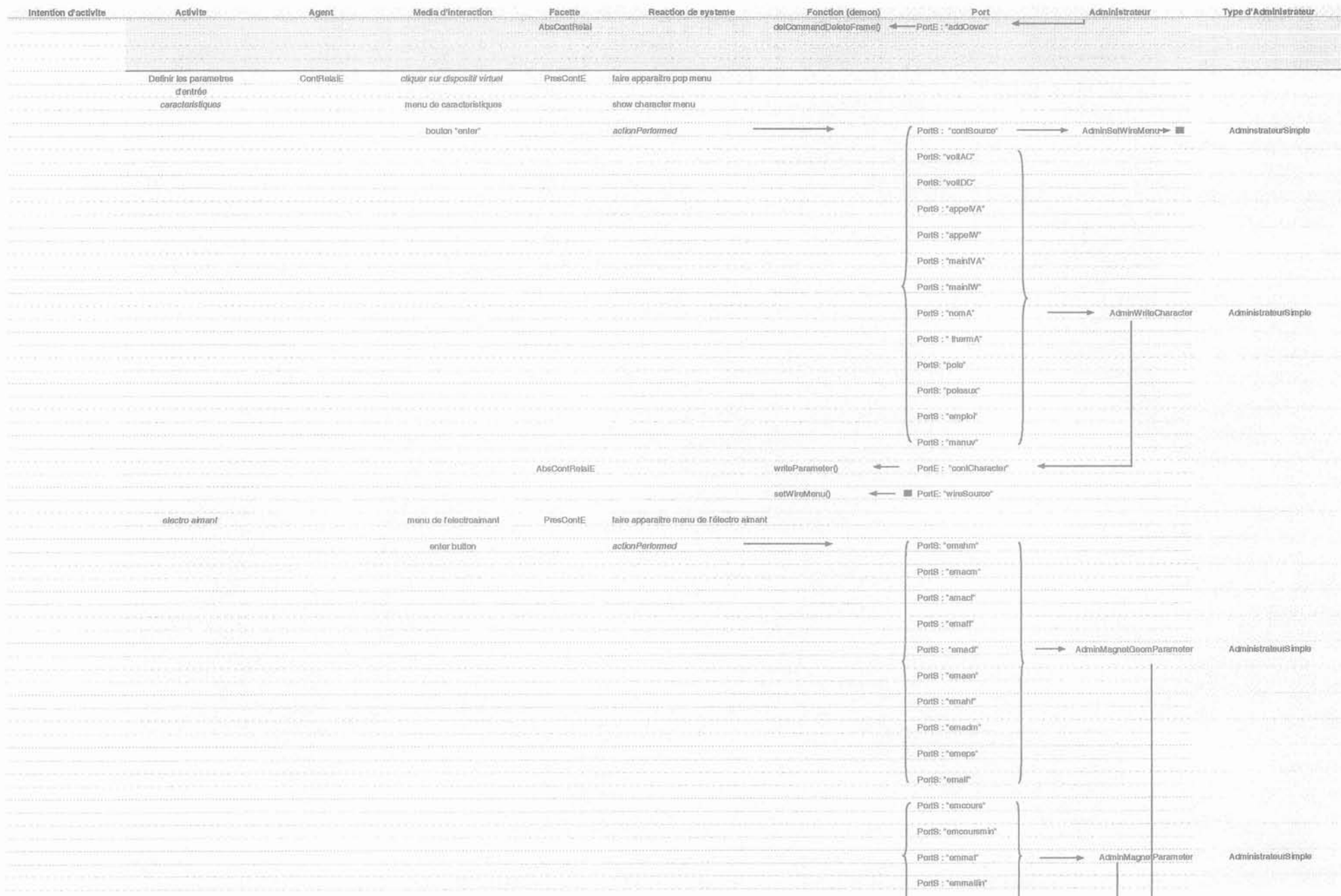
Intention d'activité	Activité	Agent	Media d'interaction	Facette	Reaction de systeme	Fonction (demon)	Port	Administrateur	Type d'Administrateur
machine AO	bouton de circuit d'excitation	PresCupboard	AbsCupboard	actionPerformed		writeExcitation()	PortE : "machExciter"		AdministrateurSimple
							Ports : "circuitExciter"	AdminExcCirtMach	
						writeExcitationCircuit()	PortE : "circuitExciter"		
							Ports (exp) : "nomDisp"	AdminNomDisp	
	bouton de fonctionnement	PresCupboard	AbsCupboard	actionPerformed		writeDeviceName()	PortE : "nomDisp"		AdministrateurSimple
	tiroir virtuel de machine AO	PresCupboard		ouvrir tiroir "MachineAO" (via BehaveMachineAODrawer)					
	tiroir virtuel de machine synchrone			ouvrir menu dialogue (via BehaveMachineACSubDrawer)			PortS : "catDisp"	AdminCatDisp	AdministrateurSimple
	bouton d'aimant	PresCupboard	AbsCupboard	actionPerformed		writeDeviceCategory()	PortE : "catDisp"		AdministrateurSimple
							Ports : "machMagnet"	AdminMagnetMach	
						writeMagnetType()	PortE : "machMagnet"		
							Ports (exp) : "nomDisp"	AdminNomDisp	
appareil	bouton de fonctionnement	PresCupboard	AbsCupboard	actionPerformed		writeDeviceName()	PortE : "nomDisp"		AdministrateurSimple
	tiroir virtuel de machine asynchrone			ouvrir menu dialogue (via BehaveMachineACSubDrawer)			PortS : "catDisp"	AdminCatDisp	AdministrateurSimple
	fonction button	PresCupboard	AbsCupboard	actionPerformed		writeDeviceCategory()	PortE : "catDisp"		AdministrateurSimple
							Ports (exp) : "nomDisp"	AdminNomDisp	
						writeDeviceName()	PortE : "nomDisp"		
créer dispositif électrique virtuel		VELaboratory	PresVELab			createSchema() (FunctionCreateSchema)	PortES : "crSchema"	AdminCreateSchema	AdministrateurSimple
								AdminCommandSchema	AdministrateurSimple
						writeCommandCreateSchema()	PortES : "crSchema"		AdministrateurSimple
								AdminCommandDevice	
	Experiment	AbsExp				writeCommandCreateDevice()	PortE (exp) : "crDevice"		AdministrateurSimple
							Ports (exp) : "catDevice"		
appareillage	Device	Device	AbsDevice			readDeviceCategory()	PortES : "catDevice"	AdminReadCatDevice	AdministrateurSimple
									AdministrateurSimple
						createDevice() (FunctionCreateDevice)	PortE : "createDevice"	AdminCreateDevice	
	contacteur	Contacteur	AbsContacteur			readDeviceName()	PortES : "catContacteur"	AdminReadCatContacteur	AdministrateurSimple
									AdministrateurSimple
						createContacteur() (FunctionCreateContacteur)	PortE : "createConf"	AdminCreateContacteur	
transformateur	Transformator	Transformator	AbsTrafo			readDeviceName()	PortES : "catTrafo"	AdminReadCatTrafo	AdministrateurSimple
									AdministrateurSimple
						createTrafo() (FunctionCreateTrafo)	PortE : "createTrafo"	AdminCreateTrafo	
							Ports : "compTrafo"	AdminReadCompTrafo	
	TransformatorNormal	TransformatorNormal	AbsTrafoTwo			readPhase()	PortES (exp) : "phaseN"		AdministrateurSequenceCible

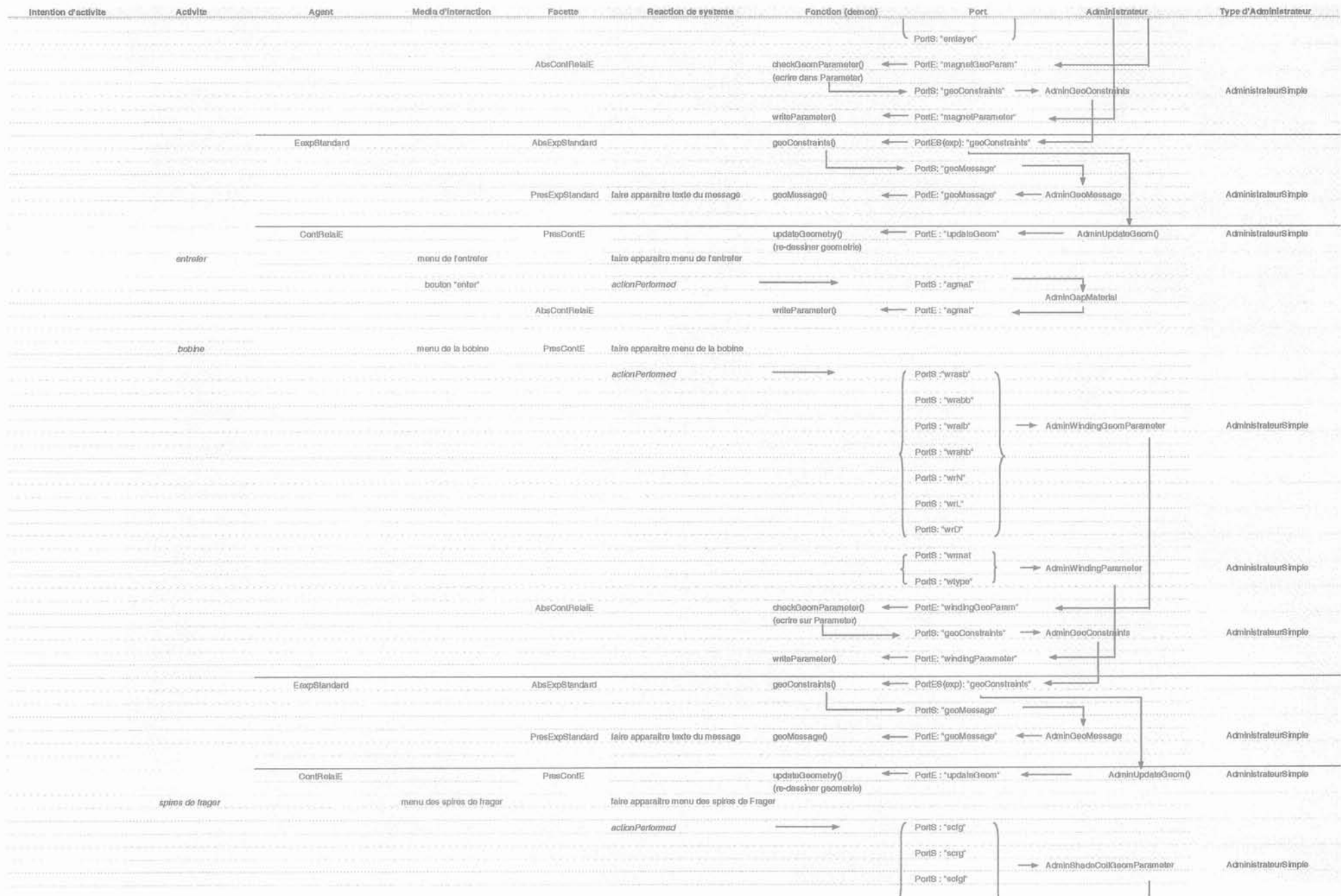
Intention d'activité	Activité	Agent	Media d'interaction	Facette	Reaction de systeme	Port	Administrateur	Type d'Administrateur
				PresTrafoTwo	createPhase() (FunctionTrafoTwoPhase)	PortE : "createPhase"	AdminCreatePhase2	AdministrateurSimple
				AbsTrafoTwo	readConnection()	PortES (exp): "coupleN"	AdminCreateCouple2	AdministrateurSimple
				PresTrafoTwo	createConnection() (FunctionTrafoTwoCouplage)	PortE : "createCouple"	AdminCreateCouple2	AdministrateurSimple
		AutoTransformer		AbsAutoTrafo	readPhase()	PortES (exp): "phaseA"	AdminCreatePhaseAuto	AdministrateurSimple
				PresAutoTrafo	createPhase() (FunctionAutoTrafoPhase)	PortE : "createPhase"	AdminCreatePhaseAuto	AdministrateurSimple
		TransformatorThree		AbsTrafoThree	readPhase()	PortES (exp): "phaseT"	AdminCreatePhase3	AdministrateurSimple
				PresTrafoThree	createPhase() (FunctionTrafoThreePhase)	PortE : "createPhase"	AdminCreatePhase3	AdministrateurSimple
				AbsTrafoThree	readConnection()	PortES : "coupleT"	AdminCreateCouple3	AdministrateurSimple
				PresTrafoThree	createConnection() (FunctionTrafoThreeCouplage)	PortE : "createCouple"	AdminCreateCouple3	AdministrateurSimple
							AdminReadCalMachDC	AdministrateurSimple
	machine DC	MachineDC		AbsMachDC	readDeviceName()	PortES : "calMachDC"	AdminReadCalMachDC	AdministrateurSimple
				PresMachDC	createMachineDC() (FunctionCreateMachDC)	PortE : "createMachDC"	AdminCreateMachDC	AdministrateurSimple
						PortS "compMachDC"	AdminReadCompMachDC	AdministrateurSequenceOble
		GeneratorDC		AbsGenDC	readMagnet()	PortES (exp): "magneGen"	AdminCreateMagnetGenDC	AdministrateurSimple
				PresGenDC	createMagnet() (FunctionGenDCMagnetCircuit)	PortE : "createMagnet"	AdminCreateMagnetGenDC	AdministrateurSimple
				AbsGenDC	readExcitation()	PortES (exp): "excitationGen"	AdminCreateExcitGenDC	AdministrateurSimple
				PresGenDC	createExcitation() (FunctionGenDCExcitation)	PortE : "createExcitation"	AdminCreateExcitGenDC	AdministrateurSimple
				AbsGenDC	readCircuitExcitation()	PortES (exp): "circuitExcitGen"	AdminCreateCircExcitGenDC	AdministrateurSimple
				PresGenDC	createCircuitExcitation() (FunctionGenDCPosExcitation)	PortE : "createCircuitExcit"	AdminCreateCircExcitGenDC	AdministrateurSimple
		MotorDC		AbsMotDC	readMagnet()	PortES (exp): "magnetMot"	AdminCreateMagnetMotDC	AdministrateurSimple
				PresMotDC	createMagnet() (FunctionMotDCMagnetCircuit)	PortE : "createMagnet"	AdminCreateMagnetMotDC	AdministrateurSimple
				AbsMotDC	readExcitation()	PortES (exp): "excitationMot"	AdminCreateExcitMotDC	AdministrateurSimple
				PresMotDC	createExcitation() (FunctionMotDCExcitation)	PortE : "createExcitation"	AdminCreateExcitMotDC	AdministrateurSimple
				AbsMotDC	readCircuitExcitation()	PortES (exp): "circuitExcitMot"	AdminCreateCircExcitMotDC	AdministrateurSimple
				PresMotDC	createCircuitExcitation() (FunctionMotDCPosExcitation)	PortE : "createCircuitExcit"	AdminCreateCircExcitMotDC	AdministrateurSimple
		MotorStepDC		AbsMotStepDC	readMagnet()	PortES (exp): "magnetStep"	AdminCreateMagnetMotStepDC	AdministrateurSimple
				PresMotStepDC	createMagnet() (FunctionMotStepDCMagnetCircuit)	PortE : "createMagnet"	AdminCreateMagnetMotStepDC	AdministrateurSimple
				AbsMotStepDC	readExcitation()	PortES (exp): "excitationStop"		AdministrateurSimple



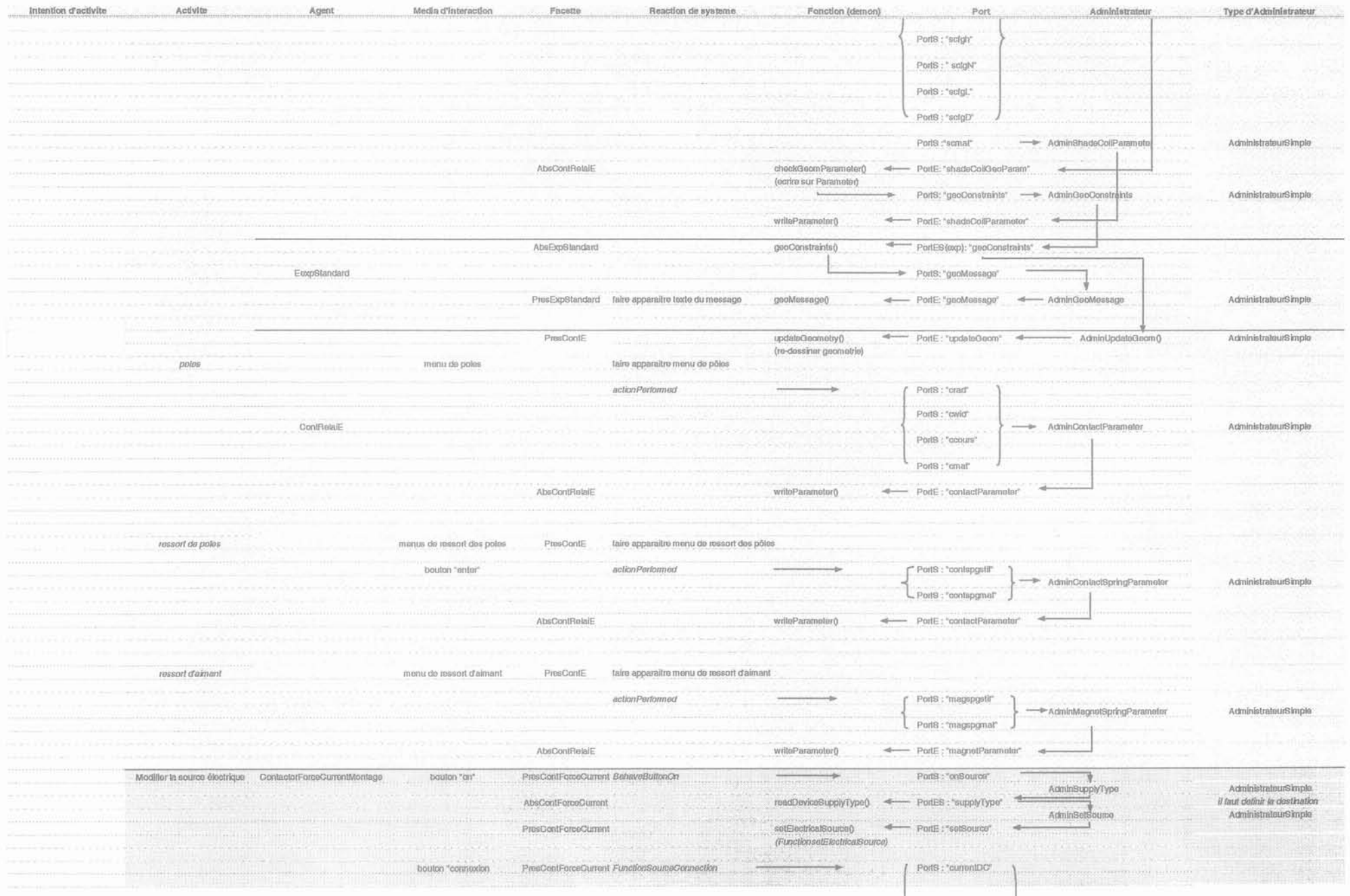


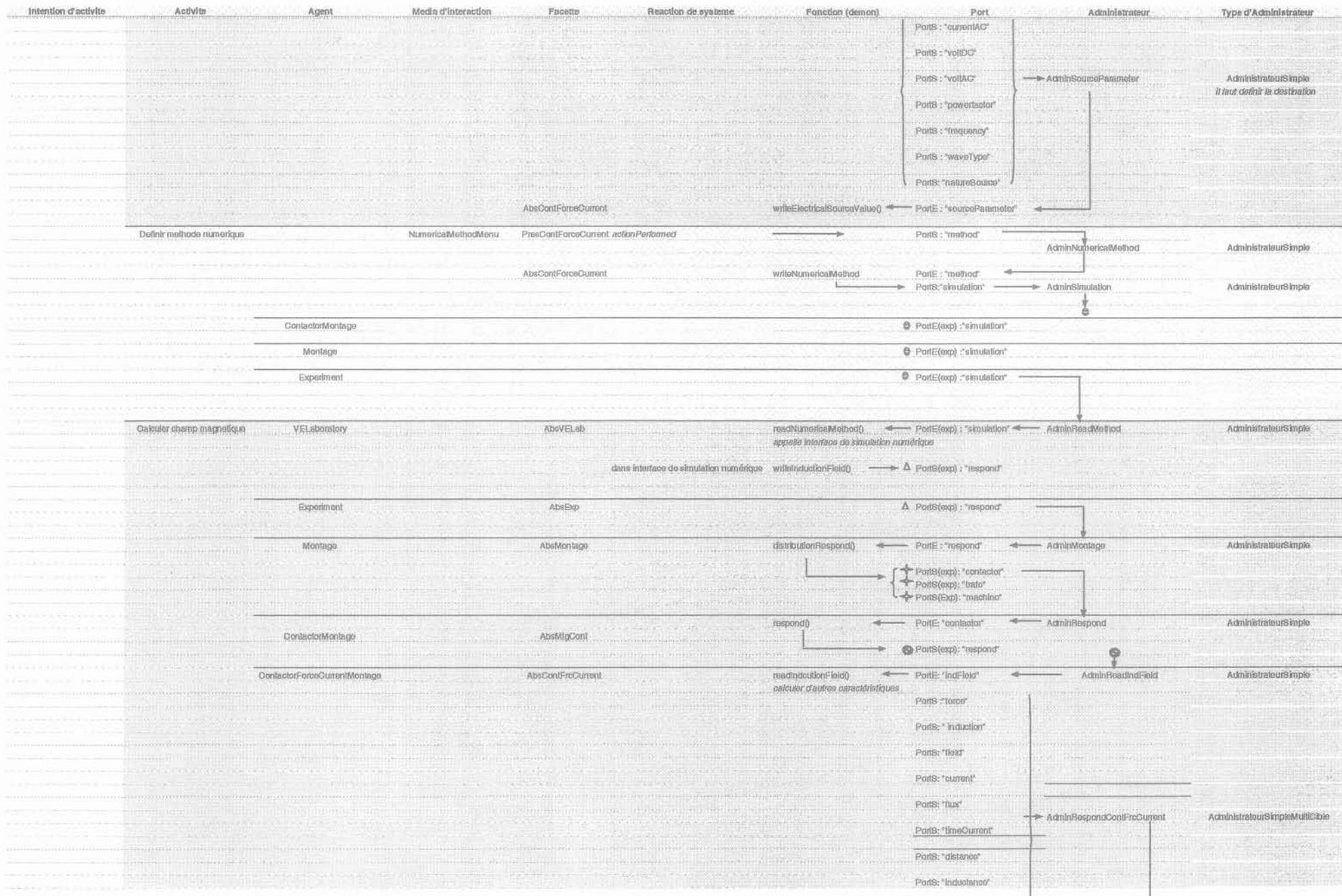






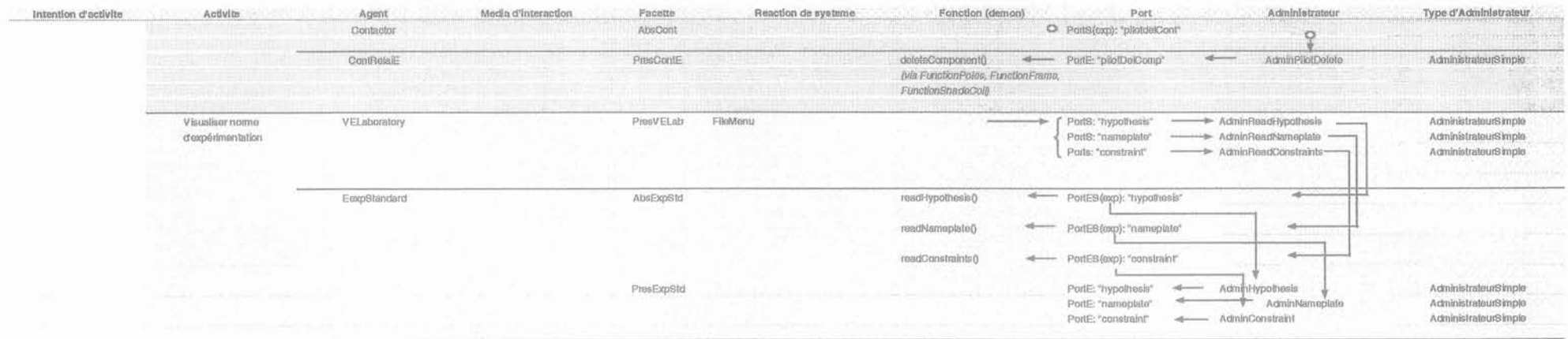






Intention d'activité	Activité	Agent	Media d'interaction	Facette	Reaction de système	Fonction (demon)	Port	Administrateur	Type d'Administrateur
							PortS: "resistant"		
							PortS: "volt"		
							PortS: "forcepulse"		
							PortS: "timeAmature"		
							PortS: "timeAmatureExp" → AdminTimeAmature	◊	AdministrateurSimple
							PortS: "eddyAllure" → AdminEddyAllure	◊	AdministrateurSimple
							PortS: "speed" → AdminSpeed	◊	AdministrateurSimple
							PortS: "animation" → AdminAnimation	◊	AdministrateurSimple
							PortS: "visualisation" → AdminVisualisation		AdministrateurSimple
	Observer les résultats			PresContFicCurrent		visualisation()	← PortE: "visualisation"		
						updateForce()	← PortE: "force"		
						updateInduction()	← PortE: "induction"		
						updateField()	← PortE: "field"		
						updateCurrent()	← PortE: "current"		
						updateFlux()	← PortE: "flux"		
						updateTimeCurrent()	← PortE: "timeCurrent"		
						updateMovement()	← PortE: "distance"		
						updateInductance()	← PortE: "inductance"		
						updateAmatureTime()	← PortE: "timeAmature"		
						updateResistance()	← PortE: "resistant"		
						updateVoltage()	← PortE: "volt"		
						updateForcePulse()	← PortE: "forcepulse"		
	Montage						PortE(exp): "animation"		
							PortE(exp): "speed"		
							PortE(exp): "timeAmature"		
							PortE(exp): "eddyAllure"		
	Experiment						PortE(exp): "animation"		
							PortE(exp): "speed"		
							PortE(exp): "timeAmature"		
							PortE(exp): "eddyAllure"		
	Device						PortE(exp): "animation"		
							PortE(exp): "speed"		
							PortE(exp): "timeAmature"		
							PortE(exp): "eddyAllure"		
	Contactor						◊ PortE(exp): "animation"		
							◊ PortE(exp): "speed"		
							◊ PortE(exp): "timeAmature"		
							◊ PortE(exp): "eddyAllure"		





## ANNEXE O : DIRECTIVE DE L'UTILISATION DE LVEE

Etapas	Explications
<p><b>Commencer une nouvelle expérimentation (en choisissant « Nouveau » sous le menu « Expérience » dans la fenêtre de la salle d'expérience)</b></p>	
<p><i>Approvisionnement dans l'armoire</i></p>	
<p>1. Choisir le dispositif étudié</p>	<ul style="list-style-type: none"> <li>- Les dispositifs électriques sont classés en trois catégories :               <ul style="list-style-type: none"> <li>• Contacteur</li> <li>• Transformateur</li> <li>• Machine tournant</li> </ul> </li> <li>- Chaque catégorie contient plusieurs types de dispositifs</li> </ul>
<p>2. Déterminer l'objectif d'expérimentation</p>	<ul style="list-style-type: none"> <li>- Les objectifs d'expérimentation sont classés en suivant les catégories des dispositifs électriques.</li> <li>- Un objectif d'expérimentation représente un schéma de montage (montage d'expérimentation)</li> <li>- Le protocole d'expérimentation apparaît en même temps que la visualisation du schéma de montage</li> </ul>
<p>3. Définir les conditions de fonctionnement du dispositif étudié</p>	<ul style="list-style-type: none"> <li>- Les conditions de fonctionnement concernent le régime et la charge. Les choix possibles de régime et de charge sont déterminés par la norme d'expérimentation.</li> <li>- Les choix de régime sont :               <ul style="list-style-type: none"> <li>• Permanent</li> <li>• Transitoire</li> </ul> </li> <li>- Les choix de charge sont :               <ul style="list-style-type: none"> <li>• Court circuit</li> <li>• Pas de charge</li> <li>• Charge</li> </ul> </li> <li>- Si le dispositif étudié a une charge, la spécification de cette charge est définie par la norme d'expérimentation</li> </ul>
<p><i>Expérimentation dans la salle d'expérience</i></p>	
<p>4. Définir les caractéristiques de fonctionnement du dispositif, et, pour chaque composant, la géométrie, le matériau et les contraintes.</p>	<ul style="list-style-type: none"> <li>- La définition doit commencer par les caractéristiques de fonctionnement et se terminer par les paramètres de chaque composant du dispositif étudié.</li> <li>- L'ordre de la définition de ces paramètres est défini par un pop menu</li> </ul>
<p>5. Mettre en œuvre la source électrique en appuyant le bouton «On»</p>	<ul style="list-style-type: none"> <li>- En utilisant les caractéristiques des paramètres de fonctionnement définies précédemment, la source est ajustée automatiquement pour être source de courant continu ou/et de courant alternatif.</li> </ul>



Etapas	Explications
6. Mettre en place la valeur de la tension ou du courant d'alimentation du dispositif par le petit bouton au-dessous de chaque écran	
7. Connecter la source au dispositif par le bouton « Connexion »	
8. Choisir la méthode numérique pour calculer le <b>champ magnétique</b> et l' <b>induction</b> (ce qui correspond au logiciel qui va exécuter la simulation numérique)	<ul style="list-style-type: none"> <li>- Chaque méthode numérique a besoin de quelques paramètres, comme par exemple : les précisions, le nombre d'itérations etc.</li> <li>- La simulation numérique est effectuée par le logiciel de simulation numérique ainsi choisi</li> </ul>
9. Observer les performances du dispositif	<ul style="list-style-type: none"> <li>- La performance physique qui correspond à la réponse principale de l'objectif d'expérimentation est montrée sur l'instrument de mesure existant</li> <li>- Un menu propose une liste des autres performances physiques qui peuvent être aussi observées. Ce menu contient un bouton pour décrire les relations entre les performances observées en utilisant des courbes</li> <li>- Un <b>navigateur</b> au-dessous de la salle d'expérience est créé pour se rapprocher du dispositif et des instruments de mesure</li> <li>- La norme d'expérimentation peut être lue en cliquant « Analyse » sur le menu barre</li> </ul>
10. Sauvegarder l'expérimentation créée	
<b>Ouvrir une expérimentation ainsi créée</b>  1. Choisir le nom de l'expérimentation créée  2. Refaire les étapes <b>Expérimentation dans la salle d'expérience</b>	

## ANNEXE P :

## 1. LE MECANISME DE CALCUL ET DE VISUALISATION DES COMPORTEMENTS PHYSIQUES D'UN CONTACTEUR ELECTROMECHANIQUE

Mécanismes (étapes)	Explication
Les étapes de la création d'un scénario d'expérimentation (un scénario de simulation) dans LVEE	
<b>(Dans un système de simulation numérique)</b> 1. Simulation numérique du fonctionnement de dispositif <i>note : calcul des phénomènes magnétiques</i>	Faite par certain système de simulation numérique pour calculer les grandeurs induction(Tesla) et champ magnétique(A/m)
<b>(Dans LVEE)</b> 2. Récupération les valeurs de l'induction et du champ magnétique	
3. Calculer le temps d'établissement du courant de la bobine	Les valeurs calculées seront visualisées sur les instruments de mesure
4. Calculer la force et le flux magnétique	
5. Calculer la vitesse de l'armature	Ces grandeurs seront utilisées pour animer le déplacement de l'armature mobile du contacteur
6. Calculer le temps de déplacement de l'armature	
Si le temps de déplacement > 0	
7. Pour $0 \leq t \leq \text{temps de déplacement}$ {	
a. Calculer entrefer (la position en cours de l'armature mobile)	
b. Calculer le courant de la bobine (pour la position en cours de l'armature mobile)	cette étape est suivie par l'activation du système de simulation numérique
c. <b>(Dans un système de la simulation numérique)</b> Simulation numérique	
d. Récupération des valeurs de l'induction et du champ magnétique	
e. Calculer la force et le flux magnétique.	
8. Calculer la température finale	Ces grandeurs sont calculées lorsque le contacteur est déjà en régime permanent mécanique ainsi qu'électrique
Si alimentation en Courant Alternatif	
9. Calculer les pertes de Foucault	
Si l'aimant est non linéaire	



Mécanismes (étapes)	Explication
10. Calculer les pertes par hystérésis	
11. Visualiser les comportements physiques ainsi calculés à travers les instruments de mesure et les courbes, et aussi sur le contacteur	

## 2. LES ROLES DES FACETTES PRESENTATION ET ABSTRACTION DE CHAQUE AGENT INTERACTIF DU MONTAGE D'EXPERIMENTATION D'UN CONTACTEUR

Objectifs d'expérimentation	Facette Abstraction	Facette Présentation
<p><b>ContactorForceCurrentMontage</b> (un montage d'expérimentation utilisé pour étudier l'influence du courant de la bobine par rapport à la force magnétique)</p> <p><i>note :</i> <i>la réponse principale :</i> la force magnétique</p> <p><i>les réponses additionnelles :</i> l'induction, le champ magnétique, le flux, et la pulsation de force (si Courant Alternatif)</p> <p><i>le régime :</i> transitoire mécanique (fermeture ou ouverture)</p> <p><i>les courbes :</i> force VS courant de la bobine, force VS entrefer, et courant de la bobine VS temps</p> <p><b>ContactorForceMasseMontage</b> (un montage d'expérimentation utilisé pour étudier l'influence du poids de la partie mobile par rapport à la force magnétique)</p> <p><i>note :</i> <i>la réponse principale :</i> idem</p> <p><i>les réponses additionnelles :</i> le courant de la bobine, l'induction, le champ magnétique, le flux et la pulsation de force (si Courant Alternatif)</p> <p><i>le régime :</i> idem</p> <p><i>les courbes :</i> idem</p> <p><b>ContactorForceRappelMontage</b> (un montage d'expérimentation utilisé pour étudier l'influence de la raideur du ressort de rappel par rapport à la force magnétique)</p> <p><i>note :</i> <i>la réponse principale :</i> idem</p> <p><i>les réponses additionnelles :</i> le courant de la bobine, l'induction, le champ magnétique, le flux et la pulsation de force (si Courant Alternatif)</p> <p><i>le régime :</i> idem</p> <p><i>les courbes :</i> idem</p>	Etape 2	
	Etape 3	recupérer la valeur du temps d'établissement du courant de la bobine
	Etape 4	
	Etape 5	note : la vitesse de l'armature est utilisée pour animer le fonctionnement du contacteur par le composant Présentation de l'agent interactif du contacteur
	Etape 6	recupérer la valeur du temps de déplacement de l'armature
	Etape 7	à la fin de la boucle, récupérer l'ensemble des valeurs de la force magnétique, l'ensemble des valeurs du flux, l'ensemble des valeurs de l'induction, l'ensemble des valeurs du champ magnétique et l'ensemble des valeurs du courant de la bobine
	Etape 9 (si Courant Alternatif)	note : l'allure du courant de Foucault est utilisée pour visualiser l'existence du courant de Foucault
	Etape 11	visualiser les comportements physiques ainsi calculés à travers les instruments de mesure et les courbes.
		note : Les valeurs visualisées sont les valeurs quand le contacteur arrive en régime permanent mécanique ainsi qu'électrique (à la fin de déplacement de l'armature mobile)

Objectifs d'expérimentation	Facette Abstraction	Facette Présentation
<b>ContactorForceResiduMontage</b> (un montage d'expérimentation utilisé pour étudier la force résiduelle)  <b>note :</b> <i>la réponse principale :</i> la force résiduelle  <i>les réponses additionnelles :</i> le flux résiduel et l'induction résiduelle  <i>le régime :</i> transitoire mécanique (ouverture)  <i>les courbes :</i> force VS courant de la bobine et force VS entrefer	Etape 2	
	Etape 3	
	Etape 4	
	Etape 5	note : la vitesse de l'armature est utilisée pour animer le fonctionnement du contacteur
	Etape 6	note : le temps de déplacement de l'armature n'est pas visualisé mais il est utilisé pour animer le déplacement de l'armature du contacteur
	Etape 7	à la fin de la boucle, récupérer l'ensemble des valeurs de la force magnétique, l'ensemble des valeurs du flux, l'ensemble des valeurs de l'induction et l'ensemble des valeurs du courant de la bobine
	Etape 11	visualiser les comportements physiques ainsi calculés à travers les instruments de mesure et les courbes .  note : Les valeurs visualisées sont les valeurs quand le contacteur arrive en régime permanent mécanique ainsi qu'électrique (à la fin de déplacement de l'armature mobile)
<b>ContactorMagnetPropertyMontage</b> (un montage d'expérimentation utilisé pour étudier les propriétés du circuit électromagnétique)  <b>note :</b> <i>la réponse principale :</i> les pertes de Foucault et ' hystérésis  <i>les réponses additionnelles :</i> la force et le courant de la bobine  <i>le régime :</i> permanent (pour certaine position de l'armature)	Etape 2	
	Etape 3	récupérer la valeur du courant de la bobine
	Etape 4	récupérer la valeur de la force magnétique
	Etape 9 (Si Courant Alternatif)	récupérer les pertes de Foucault  note : l'allure des courants de Foucault est utilisée pour visualiser l'existence de courants de Foucault
	Etape 10 (Si aimant non linéaire)	récupérer les pertes de hystérésis
	Etape 11	visualiser les comportements physiques ainsi calculés à travers les instruments de mesure et les courbes .

Objectifs d'expérimentation	Facette Abstraction	Facette Présentation
<p><b>ContacteurTimeRespondMontage</b> (un montage d'expérimentation utilisé pour étudier le temps d'ouverture ou de fermeture)</p> <p><b>note :</b> <i>la réponse principale : le temps de réponse</i></p> <p><i>les réponses additionnelles : la force mécanique, le courant d'ouverture ou de fermeture, la vitesse de l'armature. (si fermeture) le courant de maintien et le temps d'établissement du courant de la bobine</i></p> <p><i>le régime : transitoire mécanique (fermeture ou ouverture)</i></p> <p><i>les courbes : force VS courant de la bobine et force VS entrefer</i></p>	Etape 2	
	Etape 3	recupérer le temps d'établissement du courant de la bobine
	Etape 4	
	Etape 5	recupérer la vitesse de l'armature  note : la vitesse de l'armature est utilisée pour animer le déplacement de l'armature du contacteur
	Etape 6	recupérer le temps de déplacement
	Etape 7	à la fin de la boucle, récupérer l'ensembles des valeurs de la force magnétique et l'ensemble des valeurs du courant de la bobine
	Etape 9 (Si Courant Alternatif et fermeture)	note : l'allure des courants de Foucault est utilisée pour visualiser l'existence de courants de Foucault
	Etape 11	visualiser les comportements physiques ainsi calculés à travers les instruments de mesure et les courbes .  note : Les valeurs visualisées sont les valeurs quand le contacteur arrive en régime permanent mécanique ainsi qu'électrique (à la fin de déplacement de l'armature mobile)

