

UNIVERSITÉ DE MONTRÉAL

LE MODÈLE D'INFORMATION COMMUN : DESCRIPTION ET
MIGRATION DES DONNÉES CIM-XML VERS LE FORMAT EMTP-RV

AWA MARIE NDIAYE
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ECOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAITRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)

AVRIL 2007



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-29246-4

Our file Notre référence

ISBN: 978-0-494-29246-4

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

UNIVERSITÉ DE MONTRÉAL

ECOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

LE MODÈLE D'INFORMATION COMMUN : DESCRIPTION ET
MIGRATION DES DONNÉES CIM-XML VERS LE FORMAT EMTP-RV

présenté par: NDIAYE Awa Marie

en vue de l'obtention du diplôme de: Maitrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. OLIVIER Guy, Ph. D., président

M. MAHSEREDJIAN Jean, Ph.D., membre et directeur de recherche

M. ROY Gilles, M.Sc.A, membre et codirecteur de recherche

M. SAAD Omar, M.Sc.A, membre

À ma mère
À mon mari, Birame

REMERCIEMENTS

Tout d'abord, je voudrais exprimer mes remerciements et ma gratitude à mon directeur de recherche, M. Jean Mahseredjian, pour le soutien sans cesse qu'il a bien voulu m'accorder tout au long de ce projet, pour ses remarques constructives, pour ses encouragements. J'ai beaucoup apprécié sa disponibilité.

Je tiens également à remercier mon co-directeur de recherche, M. Gilles Roy, pour ses aptitudes pédagogiques, sa disponibilité et toute l'aide qu'il a pu m'apporter durant mes études de maîtrise.

Un merci bien spécial à M. Omar Saad, chercheur à l'IREQ. J'ai apprécié son intérêt pour mon travail, sa participation enthousiaste au projet, sans oublier sa bonne humeur et son ouverture d'esprit. J'ai pu compter sur M. Saad pour recevoir toute l'aide et l'appui nécessaire à la bonne réalisation du projet.

Je ne finirais pas sans exprimer toute ma reconnaissance à mon mari pour son soutien constant au cours de ces trois années de maîtrise.

RÉSUMÉ

La déréglementation de l'industrie électrique a créé le besoin pour les compagnies d'électricité, incluant les compagnies de production, les opérateurs des réseaux de distribution et de transport, d'échanger des données sur les réseaux.

Le problème de l'échange d'information entre deux ou plusieurs organisations est de mettre les données dans un format accessible et compréhensible par tous. La traduction de ces données dans un format correct pour des applications utilisant des formats propriétaires différents constitue un autre défi. La notion derrière cet échange d'information est l'interopérabilité.

Le modèle d'information commun (CIM) est né suite à un effort concerté pour faciliter l'échange d'information et l'interopérabilité des logiciels de réseaux électriques existants, quelle que soit leur source. Développé par le groupe de travail sur les interfaces des systèmes de centres de contrôle (CCAPI) de l'EPRI, le modèle de base de CIM est standardisé à travers la norme CEI 61970-301 de la Commission Électrotechnique Internationale.

Le modèle permet de représenter des objets réels qu'on peut trouver dans le fonctionnement et la gestion des réseaux électriques. Il est composé de classes pour ces objets, regroupées en paquets. Chaque classe est définie par ses attributs et par ses relations avec d'autres classes du modèle.

Pour permettre l'échange de données de réseaux, le modèle CIM a été traduit en XML en utilisant des éléments de syntaxe RDF. Ces éléments RDF permettent de décrire les instances des objets CIM et les relations entre elles. Cela conduit à l'élaboration d'un fichier XML, encore appelé document CIM-XML qui contient toutes les données de réseau à échanger.

Afin de valider ce modèle, huit tests d'interopérabilité ont été menés à ce jour et impliquaient divers vendeurs de logiciels de simulation. Le but de ces tests était de mesurer la capacité des logiciels à importer et à exporter correctement des modèles complets ou partiels de réseaux électriques sous format CIM-XML.

Dans la continuité de ces tests, ce projet fait la démonstration de la faisabilité pour le logiciel EMTP-RV d'importer des modèles de réseaux sous le format d'échange CIM-XML.

La technique de désérialisation XML est utilisée pour extraire les données contenues dans un document CIM-XML. Pour ce faire, les classes d'objets CIM et RDF ont été générées en partant des schémas XML de divers documents CIM-XML. Ces classes de types sont regroupées dans une bibliothèque de classe installée dans le registre de base. La désérialisation est effectuée dans la plate-forme .NET. Cet environnement intégré de Microsoft présente l'avantage d'offrir des outils pour la manipulation et le traitement de fichiers XML.

Une fois les données CIM-XML extraites, des correspondances ont été trouvées entre les objets CIM et les objets EMTP-RV (TLM, RLC, SM, ...). Par la suite, trois solutions ont été explorées :

- Une première solution a consisté à générer le format netlist de tous les objets CIM d'un document CIM-XML. Le fichier netlist résultant sera en fait utilisé pour lancer la simulation du réseau importé dans EMTP-RV. L'inconvénient avec cette solution, c'est que non seulement, elle ne permet pas la visualisation graphique du circuit importé mais également, il va être très délicat d'apporter des modifications futures au netlist généré comme ce dernier suit une syntaxe et une structure bien spécifiques.
- La deuxième solution a utilisé un serveur COM (activeX) créé avec .NET qui met à la disposition d'une application Javascript de EMTP-RV des méthodes retournant des objets, « clones » des objets EMTP-RV correspondant aux objets CIM. Ces objets retournés par les méthodes contiennent les données CIM-XML à importer et l'intérêt de les prendre comme des « clones » des objets EMTP-RV, c'est qu'il suffit de « mapper » ces objets avec les objets EMTP-RV pour pouvoir les représenter graphiquement dans EMTPWorks.

- Une dernière solution, plus sophistiquée, utilise la technologie des services Web. Ces derniers permettent à diverses applications de dialoguer à distance via Internet et sont basés sur des normes ouvertes telles que XML, HTTP et SOAP. Les méthodes exposées par le service Web créé sont similaires à celles de l'interface COM. La technologie AJAX est utilisée pour récupérer les données CIM et créer les émules des objets EMTP-RV comme dans la solution précédente.

Les deux dernières solutions ont permis de représenter graphiquement dans un circuit EMTPWorks les éléments CIM importés.

Mots clés : Common Information Model, Interopérabilité, Services Web, XML, RDF, plate-forme .NET, DOM, COM, AJAX, XMLHttpRequest, EMTP-RV, CEI 61970-301

ABSTRACT

The deregulation of electric industry created the need for utilities – generation, transmission, distribution – to exchange system modeling data.

The problem of information exchange between two or several utilities is to put the data to be exchanged in a format accessible and comprehensible by all. The main target behind the information exchange is interoperability.

Common Information Model (CIM) is the achievement in an attempt to facilitate information exchange and interoperation of electric utility software developed from independent sources. Developed by the working group on the Control Centre Application Programming Interface (CCAPI) of the Electric Power Research Institute (EPRI), the CIM was adopted by the IEC as standard 61970-301.

The CIM model is an abstract and object-oriented model meeting the Node/Breaker level of details needs for energy control systems operations. It specifies classes and attributes for each component of a power network, as well as the relationships between classes. Classes are gathered into packages.

In order to allow data exchange, the CIM model is translated into XML while using the RDF syntax to describe instances of CIM objects and their relationships. That results in an XML file called CIM XML document containing all the data to be exchanged.

In order to validate the CIM-XML exchange model, eight interoperation test series have been achieved. These tests target the exchange of partial, complete or incremental power system models based on the CIM-XML between different vendor products.

In the continuation of these tests, this report provides the demonstration of the capability of the EMTP-RV software to import power systems model data based on the CIM-XML representation.

The XML deserialization technique providing an easy access to the contents of an XML document is used to extract data from CIM-XML files. To make use of this technique, CIM and RDF objects types must be already created.

The framework .NET is used to carry out the majority of the operations. This integrated environment of Microsoft has the advantage of offering tools handling XML files.

The transfer of CIM data towards EMTP-RV requires finding for equivalences between CIM and EMTP-RV objects. After that, three solutions arise:

- Create the EMTP-RV netlist formats for all CIM objects in a CIM-XML document. This results in a netlist file describing the whole network to be imported into EMTP-RV. Netlist file is used for simulation. The main disadvantage of this solution is that it doesn't allow graphic visualization of the imported circuit in EMTP-RV.
- Create in the .NET framework a COM interface that allows an EMTP-RV JavaScript application to access methods returning objects that share the same attributes as EMTP-RV objects like RLC, TLM, pload The purpose is to fill the fields of created objects with CIM-XML data. The import of CIM-XML data into EMTP-RV will end by mapping the generated objects with EMTP-RV objects.
- The last solution for the data transfer towards EMTP-RV is more sophisticated and it implies the usage of Web Services. Web services are also provided by .NET and they allow remote communication between applications via Internet. They are based on open standards like SOAP, HTTP and XML.
Methods exposed through the web service are similar to those of the COM interface above. The AJAX technology allows an EMTP-RV JavaScript application to access the CIM-XML data through request and response messages.

The two last solutions made it possible to represent graphically imported power system models in an EMTP-RV circuit.

Keywords: Common Information Model, Interoperability, Web Services, XML, RDF, UML, .NET framework, data exchange, DOM, COM, AJAX, IEC 61970-301 XMLHttpRequest, EMTP-RV

TABLE DES MATIÈRES

INTRODUCTION.....	1
CHAPITRE 1 LE MODÈLE CIM.....	4
1.1 DESCRIPTION	4
1.2 LES PAQUETS DU MODÈLE CIM	6
1.3 LE LANGAGE UML – UNIFIED MODELING LANGUAGE	9
1.4 RDF – RESOURCE DESCRIPTION FRAMEWORK	12
1.4.1 <i>Description.....</i>	12
1.4.2 <i>Déclaration RDF.....</i>	12
1.4.3 <i>Graphe RDF.....</i>	13
1.4.4 <i>RDF/XML.....</i>	16
1.5 CIM/XML.....	18
1.5.1 <i>Le schéma CIM-RDF</i>	19
1.5.2 <i>Le document CIM-XML.....</i>	20
1.6 CONCLUSION	22
CHAPITRE 2 TRAITEMENT DE DOCUMENTS XML	26
2.1 ANALYSE SYNTAXIQUE.....	26
2.1.1 <i>Document Object Model</i>	27
2.1.2 <i>Limites.....</i>	28
2.2 LA SÉRIALISATION	29
CHAPITRE 3 LA PLATE-FORME .NET	30
3.1 ARCHITECTURE.....	30
3.2 ENVIRONNEMENT MULTI-LANGAGE.....	32
3.3 LES ESPACES DE NOMS.....	32
3.4 ASSEMBLAGE .NET	33
3.5 COMPOSANT COM	34
3.5.1 <i>Description.....</i>	34
3.5.2 <i>Serveur et client COM.....</i>	35
3.5.3 <i>Interopérabilité .NET et COM</i>	35
3.6 LA SÉRIALISATION XML DANS .NET	36
3.6.1 <i>Classes et interfaces.....</i>	36

3.6.2	<i>Le schéma XML</i>	37
CHAPITRE 4	EXTRACTION DES DONNÉES CIM-XML : PROCÉDURE PAS-À-PAS.....	40
4.1	CRÉATION DE LA BIBLIOTHÈQUE DE CLASSES	40
4.1.1	<i>Problématique</i>	40
4.1.2	<i>Procédure pas à pas</i>	41
4.2	ACCÈS AUX DONNÉES CIM-XML.....	44
4.2.1	<i>Désérialisation</i>	45
4.2.2	<i>Liste des types</i>	46
CHAPITRE 5	TRANSFERT DES DONNÉES CIM-XML VERS EMTP-RV	48
5.1	ARCHITECTURE DU LOGICIEL EMTP-RV	48
5.1.1	<i>Bibliothèques de composants prédéfinis</i>	51
5.1.2	<i>Programmation orientée objet</i>	51
5.1.3	<i>Simulation des circuits</i>	52
5.2	CORRESPONDANCES ENTRE OBJETS CIM ET OBJETS EMTP-RV - FORMATS NETLIST	53
5.2.1	<i>Remarques générales</i>	53
5.2.2	<i>Correspondant pour ACLineSegment</i>	54
5.2.3	<i>Correspondant pour DCLineSegment</i>	55
5.2.4	<i>Correspondant pour Breaker</i>	56
5.2.5	<i>Correspondant pour EnergyConsumer</i>	57
5.2.6	<i>Correspondant pour SynchronousMachine</i>	57
5.2.7	<i>Correspondant pour PowerTransformer</i>	59
5.2.8	<i>Correspondant pour Compensator</i>	60
5.3	RÉSULTATS.....	63
5.4	CONCLUSION	66
CHAPITRE 6	INTERFACE ACTIVEX	67
6.1	SERVEUR COM	67
6.1.1	<i>Principe de fonctionnement</i>	68
6.1.2	<i>Les méthodes</i>	70
6.2	CLIENT COM JAVASCRIPT	72
6.2.1	<i>Importation des données</i>	72
6.2.2	<i>Traitement des données</i>	73
6.2.3	<i>Résultats</i>	77
6.3	CONCLUSION	78

CHAPITRE 7	LES SERVICES WEB	80
7.1	DESCRIPTION ET CONCEPTS	80
7.1.1	<i>Architecture.....</i>	<i>81</i>
7.1.2	<i>Le protocole SOAP.....</i>	<i>83</i>
7.1.3	<i>WSDL.....</i>	<i>86</i>
7.1.4	<i>Service Web et .NET.....</i>	<i>89</i>
7.2	CRÉATION D'UN SERVICE WEB.....	89
7.2.1	<i>Déclaration de service Web</i>	<i>90</i>
7.2.2	<i>Définition des méthodes du service Web.....</i>	<i>91</i>
7.2.3	<i>Génération automatique de document WSDL.....</i>	<i>91</i>
7.2.4	<i>Test du service Web créé.....</i>	<i>93</i>
7.3	CONSOMMATION DU SERVICE WEB	94
7.3.1	<i>Technologie AJAX.....</i>	<i>95</i>
7.3.2	<i>Développement de la classe proxy</i>	<i>97</i>
7.3.3	<i>Test de la classe proxy</i>	<i>102</i>
7.4	RÉSULTATS.....	103
7.5	CONCLUSION	103
CONCLUSION		105
RÉFÉRENCES.....		109
ANNEXES		113

LISTE DES TABLEAUX

Tableau 1.1 : Objets CIM dans le modèle de réseau à 60 barres.....	25
Tableau 5.1 : Quantités d'éléments importés pour le réseau à 6 barres	63
Tableau 5.2 : Correspondances entre des éléments CIM et les objets EMTP-RV	64
Tableau 5.3 : Quantités d'éléments importés pour le réseau à 60 barres	65
Tableau 6.1 : Éléments de la bibliothèque CIM.clf.....	74

LISTE DES FIGURES

Figure 1.1: Les paquets du modèle CIM	9
Figure 1.2: Représentation en UML des classes et des différents types de relations	11
Figure 1.3: Exemple de graphe RDF à deux nœuds	14
Figure 1.4: Graphe RDF d'un petit modèle de réseau.....	15
Figure 1.5: Exemple de document XML.....	17
Figure 1.6: Exemple de correspondances entre un diagramme UML et un schéma CIM RDF ...	19
Figure 1.7: Document CIM-XML pour le diagramme de la figure 1.3	21
Figure 1.8 : Réseau test à 6 barres.....	24
Figure 2.1: Architecture d'une API XML.....	26
Figure 2.2: Objets dans la hiérarchie DOM.....	28
Figure 3.1: Le framework .NET	32
Figure 3.2: Fichier XSD du document XML de la figure 1.5	38
Figure 4.1: Fonction deser réalisant la désérialisation	45
Figure 4.2: Fonction ListTypes retournant les types d'objets.....	46
Figure 5.1 : Composant, signal et bibliothèque de composants dans EMTPWorks.....	49
Figure 5.2 : Connexion des éléments par le nom du signal.....	49
Figure 5.3 : Composant avec plusieurs niveaux de sous-circuits	50
Figure 5.4 : Correspondances dans la représentation d'un élément entre CIM et EMTP-RV	54
Figure 5.5: Netlist pour un Breaker fermé / ouvert en régime établi.....	56
Figure 6.1 : Définition d'une interface et d'une classe implémentant l'interface	69
Figure 6.2 : Code pour l'importation des charges dans EMTP-RV	76
Figure 6.3 : Résultat de l'importation du réseau res_PM.xml dans EMTP-RV	77

Figure 6.4 : Circuit de la figure 6.3 arrangé	78
Figure 7.1 : Dynamique de l'architecture d'un service Web.....	82
Figure 7.2 : Structure d'un message SOAP	85
Figure 7.3: Structure d'un document WSDL	87
Figure 7.4 : Éléments abstraits et concrets d'un document WSDL	88
Figure 7.5 : Liste des méthodes du service Web créé	92
Figure 7.6 : Test de la méthode Web ACline	93
Figure 7.7: Processus d'appel asynchrone de méthodes Web.....	98
Figure 7.8 : Exemple de format de requête SOAP 1.1	101
Figure 7.9 : Exemple de format de réponse du serveur avec le protocole SOAP 1.1.....	101

LISTE DES ACRONYMES

API : Application Program Interface
AJAX : Asynchronous JavaScript And XML
CCAPI : Central Control API
CIM : Common Information Model
EMS : Energy Management System
EPRI : Electric Power Research Institute
HTML : HyperText Markup Language
HTTP : Hyper Text Transfer Protocole
RDF : Resource Description Framework
SCADA : Supervisory Control And Data Acquisition
SGML : Standard Generalized Markup Language
SOAP : Simplified Object Access Protocol
UDDI : Universal Description, Discovery and Integration
UML : Unified Modeling Language
XML : eXtensible Markup Language
WSDL : Web Services Description Language

INTRODUCTION

Les modèles sont essentiels pour le fonctionnement des réseaux électriques modernes. La simulation des réseaux est nécessaire aussi bien pour leur planification mais également pour leur bon fonctionnement, et repose sur des modèles appropriés. La création, la maintenance et la vérification des modèles sont autant d'activités significatives pour la plupart des opérateurs de réseaux électriques.

La déréglementation de l'industrie électrique a créé le besoin pour les compagnies d'électricité, incluant les compagnies de production, les opérateurs des réseaux de distribution et de transport, d'échanger des données sur les réseaux. D'une part, l'évolution vers un libre accès aux réseaux avec un investissement dans le transport en retard par rapport à l'investissement dans la production fait que les systèmes actuels de transport de l'énergie électrique sont opérés à la limite de leurs capacités. D'autre part, dans ce contexte, une fiabilité accrue est toujours exigée. Afin de calculer de façon précise la capacité du système de transport et de gérer efficacement la congestion sur les réseaux, des modèles détaillés doivent être construits. Ces modèles, qu'on dit orientés "node/breaker"¹, sont la base sur laquelle le statut des ouvrages de transport peut être surveillé, les risques contrôlés et les transferts de puissances programmés.

Mais les modèles de réseaux qui existent aujourd'hui, dérivant des modèles de planification orientés "bus/branch"², ne sont souvent pas assez détaillés pour être adaptés aux exigences des applications des centres de conduite. Pour construire leurs modèles de réseaux détaillés et surveiller l'état du réseau, les organismes de fiabilité doivent obtenir et échanger des informations avec les utilités et les centres de conduite. Cela implique

¹ L'information requise pour l'opération en temps réel d'un réseau nécessite beaucoup de détails sur les équipements et leurs connectivités. Les modèles orientés "node-breaker" doivent par exemple, pour une sous-station représenter les différents nœuds, les éléments de commutation et les détails de mesure.

² Dans un modèle orienté "bus-branch", la représentation des équipements est simplifiée. Par exemple, les schémas de commutation et les connexions des équipements au sein d'une sous-station sont réduits. De plus, l'acquisition de données et les équipements de contrôle ne sont généralement pas modélisés.

que des modèles imbriqués ("*Overlapping models*") soient créés et coordonnés les uns avec les autres.

D'un autre côté, les données dans ces modèles peuvent être dans des formats différents, requérant ainsi la traduction des données dans le format propriétaire de la compagnie important les données.

De façon plus générale, le problème de l'échange d'information entre deux ou plusieurs organisations est de mettre les données dans un format accessible et compréhensible par tous. De plus, la traduction ou l'accès à ces données dans un format correct pour des applications utilisant des formats propriétaires constitue un autre défi. La notion derrière cet échange d'information est l'interopérabilité. Celle-ci correspond à cette capacité, plus ou moins grande, qu'ont les applications à coopérer entre elles, quels que soient leur langage, leur version, ...

Le modèle d'information commun (CIM) est né suite à un effort pour faciliter l'échange d'information entre les applications ou systèmes de gestion d'énergie (EMS) développés de façon indépendante par différents fournisseurs. Le CIM a été développé par le groupe de travail sur les interfaces des systèmes de centres de contrôle (CCAPI) de l'Electric Power Research Institute (EPRI). Le groupe CCAPI est formé tant de représentants de compagnies d'électricité que de fournisseurs de solutions progiciels. Ce groupe de travail a été chargé de définir une interface de programmation d'applications (API) qui soit normalisée pour permettre à ces applications d'échanger des informations indépendamment de la façon dont ces informations sont représentées en interne.

Le modèle CIM spécifie la sémantique de cette API. Il fournit un format commun, ouvert permettant de représenter l'information sur les réseaux électriques. Basé sur le même modèle de données, l'échange d'information peut ainsi être facilité entre les applications EMS.

Comprendre et décrire le modèle CIM, tel a été mon mandat dans le cadre d'un stage effectué à l'Institut de Recherche d'Hydro-Québec (IREQ) sous la supervision de M. Omar Saad, spécialiste du logiciel EMTP-RV. Ce projet de recherche a été naturellement le prolongement de ce stage ; l'intérêt s'est manifesté d'étudier la faisabilité que le logiciel EMTP-RV utilise le modèle CIM pour échanger des données de réseaux avec d'autres applications. En d'autres termes, il s'agit d'évaluer la compatibilité de EMTP-RV avec CIM. Cette compatibilité se traduit non seulement par une importation réussie de données sous le format CIM par le logiciel EMTP-RV mais également par le processus inverse qui consiste à transmettre sous format CIM des informations d'un circuit EMTP-RV.

Dans le cadre de ce projet, nous nous sommes limités à étudier la faisabilité de l'importation par EMTP-RV des données de réseaux sous format CIM.

Pour ce faire, les éléments d'entrée pour notre étude vont être des modèles de réseaux représentés avec le format d'échange de CIM. La première étape pour le transfert des données vers EMTP-RV est d'extraire les données de réseaux du format d'échange CIM. Cette étape est suivie par une recherche de correspondances entre les éléments du modèle CIM et les objets EMTP-RV. Pour finir, il suffit de remplir les champs des objets EMTP-RV par les données de réseaux CIM.

Ce rapport comporte trois principales parties.

La première partie décrit la structure du modèle CIM et fournit le format de son modèle d'échange (chapitre 1). Les chapitres 2 à 4 abordent les techniques pour extraire les données d'un réseau représenté en utilisant le modèle d'échange CIM. Enfin, la dernière partie (chapitres 5, 6 et 7) présente des moyens et solutions pour transférer les données CIM récupérées vers le format propriétaire EMTP-RV.

CHAPITRE 1 LE MODÈLE CIM

1.1 Description

Le modèle d'information commun (*Common Information Model* en anglais – CIM) est un modèle abstrait qui représente tous les objets principaux d'une entreprise de service public de distribution d'électricité habituellement impliqués dans les opérations de l'entreprise.

Le CIM est utilisé par les utilités, les vendeurs de logiciels de simulation et les intégrateurs pour décrire des modèles relatifs au fonctionnement des réseaux électriques. Il existe plusieurs normes basées sur le CIM, notamment les séries de normes CEI 61970 et CEI 61968 de la Commission Électrotechnique Internationale. Bien que ces normes soient en évolution, certaines parties sont déjà des normes approuvées.

C'est le cas du modèle de base CIM ("Core CIM"), standardisé à travers la norme CEI 61970-301. La première version de celle-ci a été adoptée en 2003 ; cette norme a été élaborée par le groupe de travail 13 du comité technique 57 de la CEI (IEC TC57 WG13³ en anglais) sur les interfaces de programmation d'applications pour systèmes de gestion d'énergie (EMS API).

Le modèle CIM spécifie la structure et une sémantique commune pour :

- les éléments ou ressources d'un système électrique : par exemple, une sous-station, un interrupteur, un transformateur, ...
- leurs attributs : par exemple le courant de coupure (*ampRating*) pour un disjoncteur (*Breaker*)
- et leurs relations : par exemple, un transformateur a deux ou plusieurs enroulements.

³ Ce groupe de travail fait partie du CCAPL.

CIM définit également des concepts abstraits, comme la mesure (*Measurement*) ou le contrôle (*Control*), lesquels permettent de modéliser respectivement le statut et le contrôle des données.

LE CIM est basée sur une modélisation orientée objet. Le modèle CIM définit toutes les composantes basiques d'un système électrique comme des objets. Il comprend des classes publiques et définit les attributs pour ces objets ainsi que les relations entre eux.

A cause de la taille du CIM complet, les classes d'objets qui le composent sont regroupées en un certain nombre de paquets ou ensembles (*packages* en anglais) logiques. Chaque paquet contient un ou plusieurs diagrammes de classes identifiant graphiquement toutes les classes du paquet ainsi que les relations avec les autres classes. CIM nomme systématiquement chaque classe, ses attributs et ses relations avec les autres classes, créant ainsi un dictionnaire.

Par exemple, la classe « *Measurement* » est reliée aux classes « *MeasurementType* » et « *MeasurementValue* ». Un certain nombre de « *Measurement* » peut être relié à un « *Terminal* » d'un « *ConductingEquipment* » (il faut comprendre par là que plusieurs mesures peuvent être faites aux bornes d'un équipement). Ainsi, par exemple, le courant de coupure d'un disjoncteur (*Breaker*) serait défini par :

- *Breaker.Terminal.Measurement.MeasurementType.name*
- et *Breaker.Terminal.Measurement.MeasurementValue.value*

La liste ainsi qu'une brève description des paquets du modèle de base CIM est donnée dans la section suivante.

A noter que CIM n'est ni une modélisation de base de données ni un format d'échange. Le CIM est un modèle statique, dans le sens où il ne spécifie pas comment les applications peuvent contrôler la consistance des attributs à l'intérieur des objets. Par exemple, si le statut d'un disjoncteur a changé (de *Open* à *Closed*), il est nécessaire d'obtenir un nouveau set de valeurs à partir de *MeasurementValue*.

1.2 Les paquets du modèle CIM

Le modèle CIM comporte un modèle de base, documenté dans la norme 61970-301. Ce modèle de base CIM est constitué d'un ensemble de paquets (*packages*) qui offrent une vue logique des aspects physiques des informations de l'EMS.

Ce modèle de base comporte dix paquets.

- **Le "Core Package"** (ou paquet noyau) comporte les objets de base nécessaires pour la plupart des applications implémentant le CIM. Il s'agit par exemple des classes *PowerSystemResource* ou *ConductingEquipment* :
 - La classe *PowerSystemResource* comporte toutes les composantes du système qui peuvent être soit des éléments individuels (interrupteur par exemple) ou un groupe d'éléments (sous-station par exemple).
 - La classe *ConductingElement* regroupe tous les éléments du réseau pouvant porter du courant.
- **Le "Topology Package"** fournit les ports de connectivité pour relier les objets du *Core Package* via le *Wires Package*. Trois classes existent dans ce paquet :
 - Classe *ConnectivityNode* qui correspond au nœud auquel le terminal d'un équipement est connecté de façon franche (impédance nulle).
 - Classe *TopologicalIsland* qui représente un sous-ensemble de réseau électriquement connecté.
 - Classe *TopologicalNode* modélisant un ensemble de nœuds de connectivité connectés ensemble via des interrupteurs fermés.

Il est intéressant de rappeler que le modèle CIM utilise la représentation "node-breaker"; celle-ci est à un niveau de détail plus élevé que la représentation de type "bus/branch" (de la plupart des logiciels de simulation de réseaux existants).

- **Le "Wires Package"** modélise l'information sur les équipements électriques des réseaux de distribution et de transport. Il inclut également les équipements d'un réseau de répartition ou encore d'une sous-station. Parmi les classes de ce paquet, figurent celles pour les lignes à courant alternatif / courant continu, le

conducteur, le transformateur, le redresseur/onduleur, la machine synchrone, une charge, ...

L'information fournie est utilisée pour connaître l'état du réseau (Network Status), les estimations d'états (State Estimation), pour réaliser un écoulement de puissance (Load Flow) ou encore pour des applications d'analyse de contingences (Contingency Analysis).

- Le "**Domain Package**" est un dictionnaire des quantités et des unités qui définissent les types de données utilisés par les attributs (pour toute classe définie dans le modèle CIM).
- Le "**Meas Package**" comporte les classes relatives à la mesure (unités, quantité, valeur, ...)
- Le "**Generation Package**" comporte deux sous-paquets :
 - *Production Package*
 - *GenerationDynamics Package*

Le premier sous-paquet contient les classes permettant de modéliser tout type de générateur électrique. Le sous-paquet *GenerationDynamics* regroupe les classes pour la description des "prime-movers" : types de turbine, chaudières, ...

Les informations contenues dans ce modèle peuvent être utilisées pour effectuer un dispatching économique, pour des applications de contrôle automatique de la production, ...

- Le "**Outage Package**" : les classes de ce paquet permettent de modéliser la configuration courante et future du réseau (planification des travaux sur certains équipements)
- Le "**LoadModel Package**" regroupe les classes permettant de décrire tout niveau de charge, allant d'un consommateur final à la charge d'une région. La charge peut être modélisée avec des courbes variant en fonction du temps, lesquelles représentent les effets des saisons ou du type de jour. La dépendance de la charge vis-à-vis de la tension et de la fréquence peut aussi être modélisée.

- Le "**Protection Package**" comporte toutes les classes permettant de modéliser les appareils de protection et leurs caractéristiques.

En fait, la norme 61970-301 de base fait partie de la série CEI 61970 qui définit une interface de programmation d'application (API) pour un système de gestion d'énergie (EMS). Cette série 61970 pour CIM comporte d'autres parties qui sont en état de brouillons ou en cours de préparation. Entre autres sections, il y a :

- une partie 302 (61970-302) qui porte sur les échanges de données entre compagnies. Il couvre les aspects financiers, de programmation et de réservation. Trois paquets existent pour ce modèle :
 - Paquet « **EnergyScheduling** » pour la programmation des quantités d'électricité à échanger. Les classes de ce paquet fournissent l'information sur les quantités de MW produites, consommées, perdues, transitées, achetées, vendues.
 - Paquet « **Financial** » pour tout ce qui est règlement et facturation. Les classes de ce paquet représentent les entités légales participant aux échanges.
 - Paquet « **Reservation** » contient l'information sur les capacités de génération, les services auxiliaires, le transport.
 - une partie 303 (61970-303) relative à la télésurveillance et acquisition de données (Supervisory Control And Data Acquisition – **SCADA**). Il ne contient qu'un paquet qui permet de modéliser l'information utilisée par les applications SCADA. La télésurveillance permet de commander l'équipement, comme l'ouverture ou la fermeture d'un disjoncteur. L'acquisition de données regroupe toutes les données télé-mesurées à partir de diverses sources.
- Ce modèle décrit les mesures, les transformateurs (de tension et de courant), les circuits de communication, ...

Le schéma de la figure 1.1 reprend tous ces paquets en indiquant par des traits les paquets liés. Pour plus de détails et d'informations sur les classes de ces différents paquets, se référer au document [1].

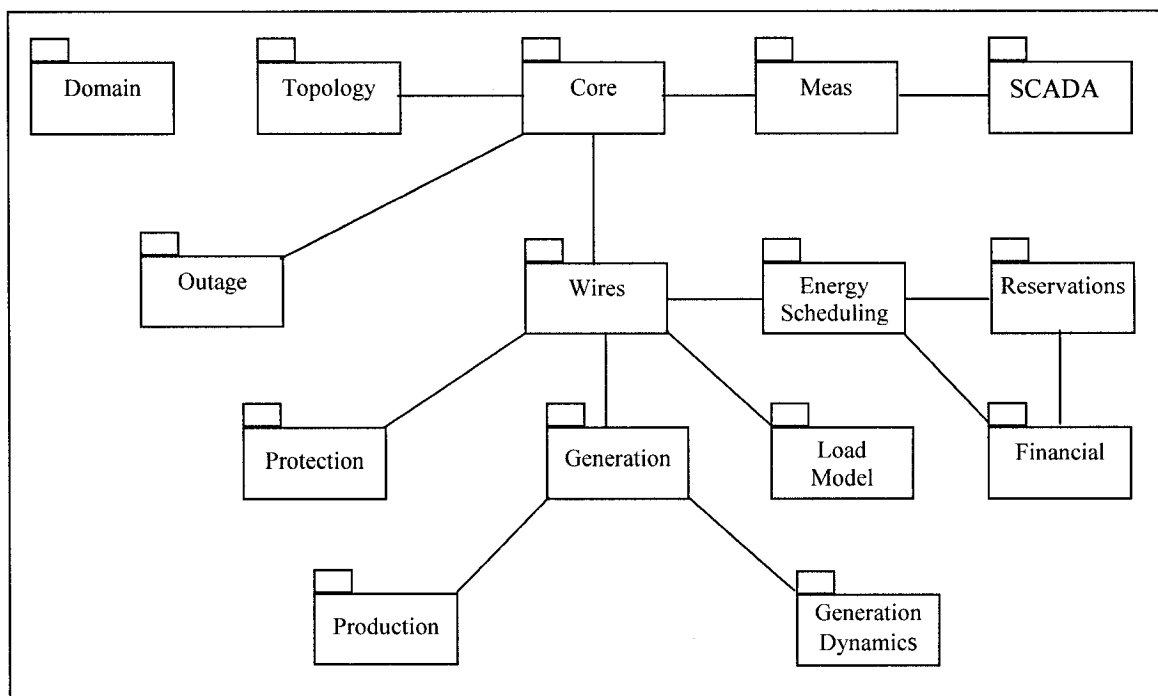


Figure 1.1: Les paquets du modèle CIM

La notation UML (Unified Modeling Language), est utilisée pour représenter la structure – les classes et les relations entre les classes – des diagrammes ou schémas CIM. Ces derniers sont fournis comme des fichiers de modèles UML Rational Rose.

1.3 Le langage UML – Unified Modeling Language

Le langage unifié de modélisation (UML) est un ensemble de concepts et de conventions graphiques servant à exprimer les modèles orientés objet de systèmes complexes. Il s'agit d'un langage de modélisation orientée objet, utilisant une représentation graphique. Il a été normalisé par l'OMG (Object Management Group) en 1997, ce qui a contribué à en faire le langage standard de l'industrie pour spécifier, visualiser, construire et documenter des modèles.

Le langage UML est utilisé pour représenter la structure des schémas CIM. Avec cette notation, les modèles de systèmes électriques sont exprimés sous forme de diagrammes de classes : tous les objets du système électrique sont modélisés par des classes, ayant des attributs et des relations entre elles.

En UML, une classe est représentée par un rectangle. Une classe contenant des attributs est représentée par un rectangle divisé en deux zones, l'une contenant le nom de la classe, l'autre la liste des propriétés. Les méthodes sont représentées par une troisième zone contenant la liste des méthodes.

Les relations entre les classes peuvent être principalement de trois types : héritage, agrégation ou association.

- Avec l'héritage, une classe (sous-classe) hérite de toutes les propriétés de la classe parente (super classe), à savoir les attributs et les méthodes.

Le symbole représentant l'héritage en UML est une flèche pointant vers la super classe. Par exemple, les classes *ConductingEquipment* et *Substation* sont des sous-classes de la super classe *PowerSystemResource*. De même, la classe *Breaker* est dérivée de la classe *Switch* qui, elle-même est une sous-classe de *ConductingEquipment* (cf. fig. 1.2). Autrement dit, un *Breaker* est un type de *Switch*, qui lui-même est un type de *ConductingEquipment*.

- L'agrégation correspond à une relation « composé-composant » où les objets représentant les composants d'une chose sont associés à un objet représentant l'assemblage (ou l'agrégation) entier. Autrement dit, une classe est construite en utilisant les instances d'une autre classe n'existant que comme dérivée de la classe parente. Le symbole représentant l'agrégation en UML est le losange vide placé du côté du composé.

Par exemple, une séquence de fermeture est définie pour toute fermeture possible du *Breaker*. Donc un *Breaker* est constitué d'une classe "Séquence de fermeture" (*RecloseSequence*). La classe *RecloseSequence* n'existe que par rapport au *Breaker* et ne peut être autonome (cf. fig. 1.2).

- Une association indique que deux classes indépendantes sont reliées, permettant ainsi un flux bidirectionnel des données, mais les classes peuvent exister l'une sans l'autre. Par exemple, les disjoncteurs (classe *Breaker*) peuvent être opérés par des relais de protection (classe *ProtectionEquipment*) ou encore un *Terminal* est connecté à un *ConnectivityNode*.

Une association est représentée en UML par une ligne sur laquelle figure, généralement au milieu, le nom de l'association.

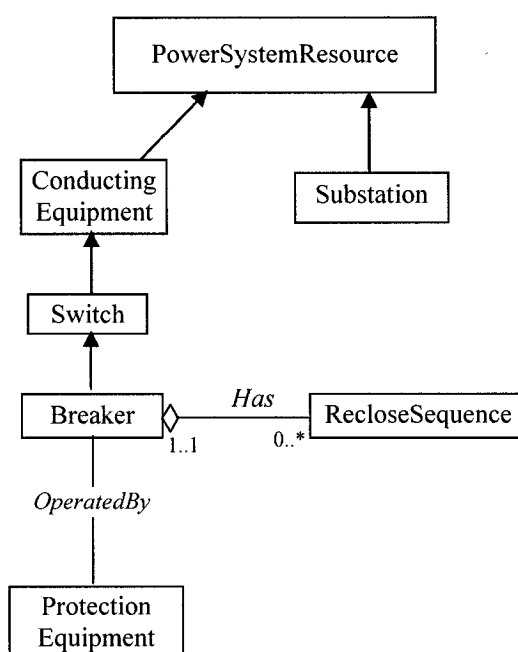


Figure 1.2: Représentation en UML des classes et des différents types de relations

L'agrégation et l'association sont des relations binaires.

Pour plus d'informations dans la représentation UML, se référer à [2].

Le langage UML, via le diagramme de classes, permet donc de se raccrocher aux concepts abstraits définis dans le modèle CIM. Maintenant, il faut faire communiquer l'information contenue dans ces diagrammes de classes.

La CEI a proposé d'utiliser un schéma RDF (Resource Description Framework) pour échanger des données dans un format commun. Le schéma RDF est documenté par la CEI dans la norme 61970-501. Le modèle RDF est décrit dans la section suivante.

1.4 RDF – Resource Description Framework

1.4.1 Description

Resource Description Framework (RDF) est un modèle conceptuel permettant de décrire simplement et de manipuler des métadonnées⁴. Il a été normalisé par le consortium W3C (Wide World Web Consortium – <http://www.w3c.org>).

RDF est un métalangage tout usage pour représenter les aspects descriptifs d'une ressource Web – comme le titre, l'auteur et la dernière date de révision d'un document Web, ... RDF offre non seulement un cadre fonctionnel pour exprimer ces informations, mais, de plus, il le fait de façon à ce que ces informations puissent être échangées entre les applications.

De façon générale, le but de RDF est de définir un mécanisme neutre de description des ressources applicable à n'importe quel domaine d'application.

RDF utilise un modèle de graphes pour décrire et permettre un certain traitement automatique des métadonnées.

1.4.2 Déclaration RDF

Le concept de base de RDF consiste à décrire une ressource en utilisant un jeu de propriétés. Chacune de ces propriétés possède un type et une valeur pouvant servir à décrire n'importe quelle ressource.

RDF décrit trois types d'objets : ressource (*resource*), propriété (*property*) et valeur (*value*).

- Une ressource est tout élément (instance d'objet) qui peut être identifié de façon unique par un URI ("Uniform Resource Identifier"). Les ressources sont regroupées en classes.

⁴ Une métadonnée est un ensemble structuré d'informations décrivant une ressource quelconque.

- Une propriété désigne un attribut, une caractéristique qui s'applique à une ressource. Il peut également s'agir d'une relation binaire entre deux ressources.
- Les valeurs sont les valeurs affectées aux propriétés ; une valeur peut être soit une ressource soit un littéral⁵ ("literal").

Une ressource spécifique associée à une propriété et sa valeur correspondante est une assertion ou déclaration (« *statement* ») RDF.

Une autre façon de représenter une déclaration RDF est de la mettre sous la forme {prédicat, sujet, objet}. Le prédicat est une propriété, le sujet est une ressource et l'objet de la déclaration correspond à la valeur de la propriété.

En plus des déclarations RDF, les spécifications RDF [4] montrent deux autres représentations d'un modèle de données :

- Une représentation graphique avec le schéma (ou graphe) RDF,
- Une représentation textuelle en utilisant la syntaxe XML.

1.4.3 Graphe RDF

1.4.3.1 Ontologie RDF Schema

Le modèle RDF sert à annoter des documents pour décrire leur contenu. Les descriptions reposent sur un modèle partagé : une ontologie. Une ontologie est un ensemble structuré de concepts. Ces concepts sont organisés dans un graphe dont les relations peuvent être sémantiques ou des relations de composition et d'héritage (source : Wikipedia.com). Cet ensemble de concepts permet de définir complètement un domaine.

Les énoncés RDF peuvent se rapporter à une ontologie du domaine, exprimée en **RDF Schema (RDFS)** [3]. Ce dernier spécifie donc le vocabulaire des descriptions RDF.

Le modèle RDF comporte un nombre de classes et de propriétés prédéfinies. Toutes les ressources sont regroupées en classes. Une classe est une ressource de type **rdfs:Class**, identifiée de manière unique et décrite par des propriétés.

⁵ Un littéral désigne un type primitif tel qu'un entier ou une chaîne de caractères.

Une propriété est une ressource de type `rdf:Property`. La propriété `rdf:type` est utilisée pour indiquer qu'une ressource est une instance d'une classe. Une propriété permet de définir une relation binaire entre une ressource « sujet » et une ressource « objet ». RDFS permet également de restreindre le domaine d'application (`rdfs:domain`) et le domaine de variation (`rdfs:range`) d'une propriété.

La relation d'héritage entre classes et celles entre propriétés est définie dans RDFS respectivement avec `rdfs:SubClassOf` et `rdfs:SubPropertyOf`.

1.4.3.2 Diagramme des nœuds et des arcs

Les graphes RDF encore appelés diagrammes des nœuds et des arcs (*nodes and arcs diagrams* en anglais) sont simplement constitués de nœuds connectés par des arcs. C'est en fait un ensemble de triplets.

- Les nœuds dans un graphe RDF, dessinés sous forme d'ovales, représentent les ressources et peuvent être de trois types : URI, vide et littéral. Un nœud URI signifie « la ressource indiquée par l'URI », un nœud vide signifie « une ressource quelconque ». Les nœuds représentant des « chaînes de caractères » (*literal*) sont représentés par des rectangles.
- Les arcs représentent les noms des propriétés. Un arc part toujours du sujet (la ressource) et pointe vers l'objet.

Le graphe de la figure 1.3 montre deux nœuds représentés par des ovales, reliés par un seul arc ; un premier nœud est étiqueté « Relais 1A » et l'autre « CB 101 ». L'arc connectant ces deux nœuds porte l'étiquette « Operates ».

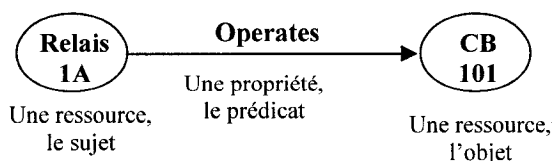


Figure 1.3: Exemple de graphe RDF à deux nœuds

Une façon de traduire ce graphe consiste à dire que "le relais 1A opère/fait fonctionner le disjoncteur CB 101".

La figure 1.4 montre un ensemble de déclarations représentant un petit modèle. Il fait état d'un relais (relais 1A) faisant fonctionner un disjoncteur (CB 101). Celui-ci a sa borne 101B (Terminal 101B) connectée à la borne 21 (Terminal 21) de l'enroulement 21 du transformateur 1 (T1-enroul. 21). Le disjoncteur 101 est un élément de la Baie 1, qui elle-même appartient à la compagnie Société.

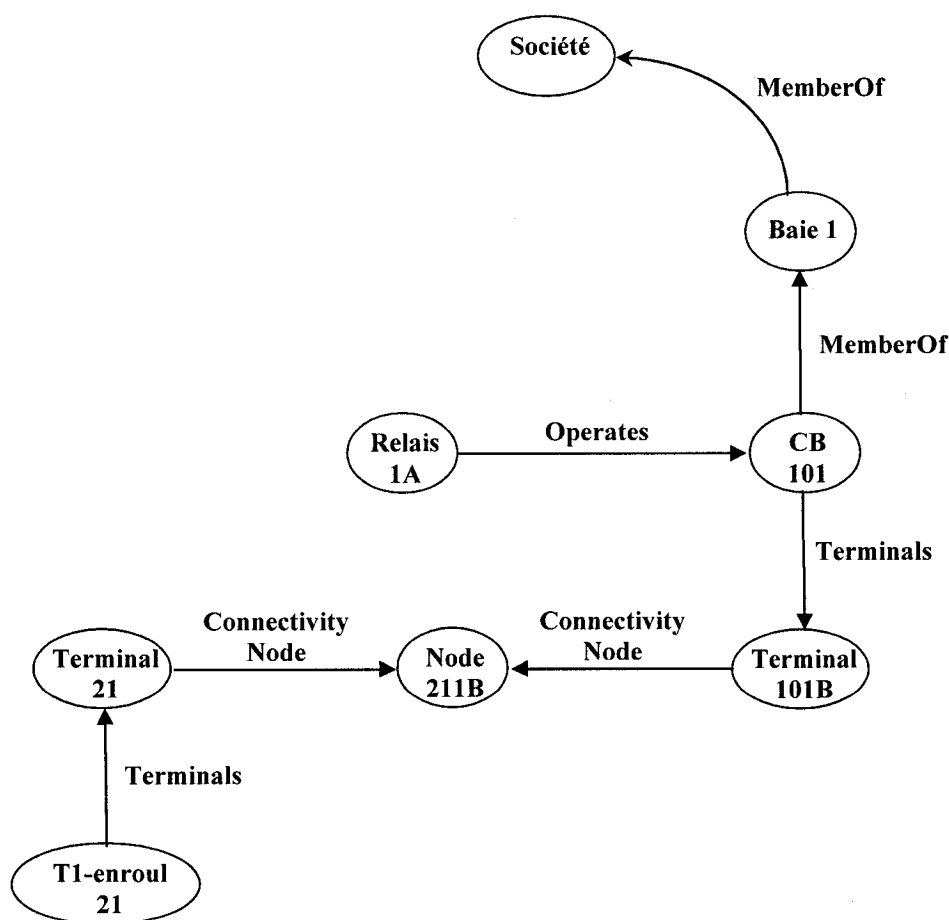


Figure 1.4: Graphe RDF d'un petit modèle de réseau

1.4.4 RDF/XML

Afin de créer et de pouvoir échanger ces métadonnées, une syntaxe concrète est requise pour spécifier les triplets. RDF/XML⁶ est un des formats de sérialisation de RDF. Il utilise le langage XML ("eXtensible Markup Language").

1.4.4.1 Le format XML

Le langage XML [5] est un langage de balisage développé par le consortium W3C.

XML est un format texte simple et flexible, dérivant du SGML (Standard Generalized Markup Language) et de l'HTML (HyperText Markup Language). Tout comme HTML, XML est une norme régissant la mise en forme d'un contenu en utilisant des balises.

A la différence du langage HTML, un utilisateur peut créer ses propres balises en XML (balises redéfinissables). De plus, la définition des balises et leurs relations sont spécifiées dans une DTD (*Document Type Definition*) ou un schéma XML.

Aussi, ce format est très populaire pour l'échange de données à travers Internet ou pour des réseaux privés.

Un document XML est composé d'une définition de sa structure et d'un contenu. La structure d'un document XML commence par un prologue et est souvent représentée graphiquement comme un arbre. Cet arbre est constitué d'éléments imbriqués les uns dans les autres (ayant une relation parent-enfant) et d'éléments adjacents.

- La première ligne forme le prologue, constitué d'une déclaration XML définissant la version XML et le jeu de codage de caractères utilisé dans le document, comme le montre la figure 1.5.
- L'élément RESEAU est l'élément racine (*document element* en anglais) qui est unique. Il constitue la base du document XML
- Les éléments définis à l'intérieur de l'élément racine sont les éléments enfants (ou éléments principaux) de RESEAU. Ces éléments (zone, Lien) forment la

⁶Bien que RDF/XML ne soit qu'une sérialisation du modèle, elle est souvent appelée RDF. Un abus de langage désigne à la fois le graphe de triplets et la présentation XML qui lui est associée.

structure même du document : ce sont les branches et les feuilles de l'arbre. Ils peuvent contenir du texte, ou bien d'autres éléments (sous-éléments). Par exemple, l'élément "zone" contient les sous-éléments "Charge" et "Prod".

- Tous les éléments peuvent contenir un ou plusieurs attributs. Un attribut est composé d'un nom et d'une valeur. Par exemple, l'élément "zone" possède un attribut "nom".

```
<?xml version="1.0" encoding="utf-8"?>
<RESEAU>
  <zone nom="Zone A">
    <Charge>10000MW</Charge>
    <Prod>9000MW</Prod>
  </zone>
  <zone nom="Zone B">
    <Charge>20000MW</Charge>
    <Prod>21000MW</Prod>
  </zone>
  <Lien nom="Interconnexion">
    <Ptransit>1000MW</Ptransit>
  </Lien>
</RESEAU>
```

Figure 1.5: Exemple de document XML

Avec la structure d'un document XML représentée sous forme d'arbre, XML présente l'avantage d'être souple et extensible.

Un certain nombre de règles de base doivent être respectées lors de l'élaboration d'un document XML :

- Un nom d'élément ne peut commencer par un chiffre. Si le nom n'est composé que d'un seul caractère, ce doit être une lettre comprise entre "a" et "z" pour les minuscules, "A" et "Z" pour les majuscules. S'il est composé d'au moins deux caractères, le premier peut être "_" ou ":". Le nom peut ensuite être composé de lettres, chiffres, tirets, tirets bas et deux points. La syntaxe XML est sensible à la casse (le format distingue majuscules et minuscules).

- Toutes les balises portant un contenu non vide doivent être fermées. La balise de début (<zone>), la balise de terminaison (</zone>) et le contenu entre deux sont globalement appelés *élément*.
- Les balises n'ayant pas de contenu doivent se terminer par />.
- Les noms d'attributs sont en minuscules. Les valeurs d'attributs doivent être entre guillemets.

Un document respectant ces critères est dit bien formé (*well formed* en anglais).

1.4.4.2 Structure du document RDF/XML

RDF/XML est organisé sous la forme d'un oignon où les éléments de niveau 1 (à l'intérieur de l'élément racine **rdf:RDF**) représentent des ressources dans le modèle RDF, les éléments de niveau 2 représentent des propriétés de ces ressources (des arcs dans le graphe RDF), les éléments de niveau 3 représentent les ressources qui sont les valeurs de propriétés et ainsi de suite.

La structure du graphe est codée dans un arbre XML grâce à l'utilisation des attributs **rdf:about**, **rdf:ID**, **rdf:resource**, ... (cf. référence [3]).

Un document RDF fait référence à au moins deux espaces de noms :

- *rdf* d'une part,
- d'autre part le domaine auquel appartiennent les données.

L'espace de nom est donné par l'attribut **xmlns**.

L'espace de nom **rdf** est désigné par l'URI : "**http://w3.org/1999/02/22-rdf-syntax-ns#**". Utilisé avec la sérialisation RDF/XML, cet URI correspond au nom d'espace associé avec le vocabulaire RDF (RDFS).

1.5 CIM/XML

CIM-XML est une application du RDF au modèle d'information commun (CIM) de l'EPRI, donc adapté aux systèmes électriques.

1.5.1 Le schéma CIM-RDF

Un schéma CIM-RDF est obtenu en **codant le modèle abstrait CIM avec le vocabulaire du RDFS**.

Comme nous l'avons vu plus tôt, RDF est assez global pour pouvoir décrire les concepts UML. D'ailleurs la note [6] décrit les relations entre le schéma RDF et UML. La figure ci-dessous illustre quelques-unes des correspondances.

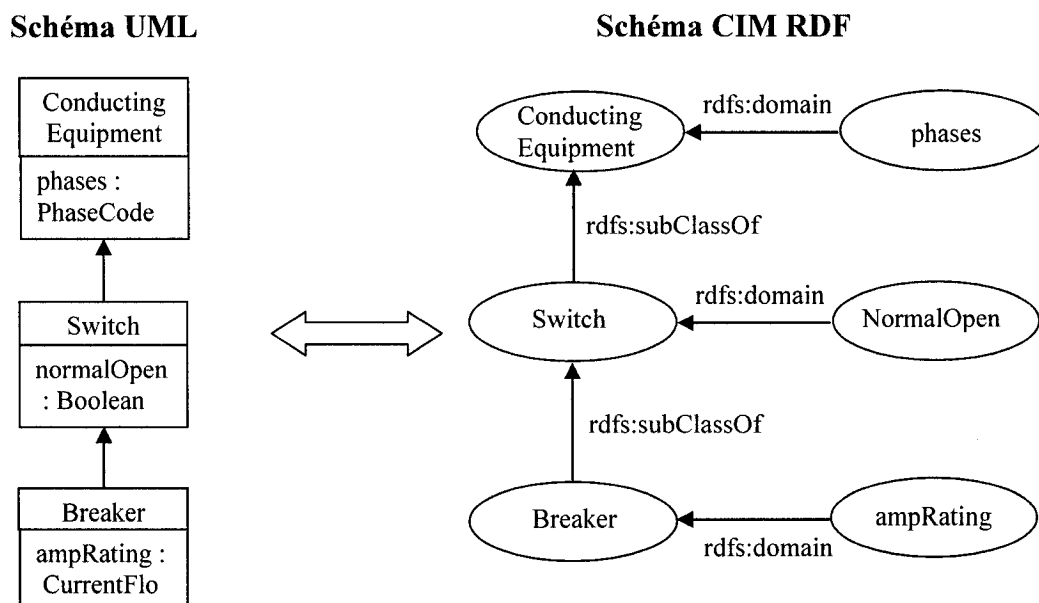


Figure 1.6: Exemple de correspondances entre un diagramme UML et un schéma CIM RDF

Les ressources UML correspondent aux objets RDF : les classes UML sont traduites en classes RDF. Les attributs des classes UML sont traduits en RDF en des classes de propriétés ("phases", "NormalOpen", "ampRating") reliées aux classes de ressources par "rdfs:domain". Les relations d'héritage UML sont traduites en RDF avec la classe de propriété "SubClassOf".

La norme CEI 61970-501 décrit les mécanismes pour convertir un modèle UML en un schéma RDF. La conversion du modèle UML dans les fichiers Rational Rose peut se

faire manuellement ou supporté par l'outil de conversion Xpetal⁷. Les fichiers en entrée de l'outil Xpetal sont des fichiers .mdl ou .ptl (suivant la version de Rational Rose - 4 ou 98) et le fichier de sortie est un schéma RDF. L'outil Xpetal a été développé par les consultants Langdale (www.langdale.com.au).

1.5.2 Le document CIM-XML

Après avoir créé le schéma CIM-RDF, le modèle de réseau peut alors être converti pour être exporté sous la forme d'un document XML. Toutes les ressources présentes dans le graphe RDF sont décrites en utilisant la syntaxe XML.

Mais la quantité de données échangées entre des systèmes de gestion d'énergie (EMS) ou même entre compagnies peut être très importante, et traiter ces données peut consommer beaucoup de temps, comme les « tags » des éléments sont imbriqués entre eux.

Aussi, la CEI a proposé un format d'échange CIM-XML simplifié [19]. Celui-ci utilise une syntaxe simplifiée de RDF [8]. Cela rend plus facile, entre autres, la construction de scripts de sérialisation et de désérialisation. Avec cette représentation simplifiée, une relation d'association ou d'agrégation entre deux ressources (éléments CIM) est écrite en utilisant l'attribut « ressource » (**rdf:resource**) au lieu d'imbriquer un élément dans l'autre.

Le fichier XML, sous cette représentation CIM-XML simplifiée, est appelé document CIM-XML. Cette syntaxe simplifiée a été adoptée par le NERC (North American Electric Reliability Council) et a fait l'objet de la norme CEI 61970-503.

L'exemple de la figure 1.7 montre le document CIM-XML du schéma de la figure 1.4.

⁷ Disponible seulement sur le site de Langdale Consultants au <http://www.langdale.com.au/styler/xpetal>

```

<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns# "
  xmlns:cim="http://iec.ch/TC57/2000/CIM-schema-cim10#">
  <cim:Company rdf:ID="ID1">
    <cim:Naming.name>Societe</cim:Naming.name>
  </cim:Company>
  <cim:Bay rdf:ID="ID2">
    <cim:Naming.name>Baie1</cim:Naming.name>
    <cim:PowerSystemResource.MemberOfCompany rdf:resource="#ID1"/>
  </cim:Bay>
  <cim:Breaker rdf:ID="ID3">
    <cim:Naming.name>CB101</cim:Naming.name>
    <cim:PowerSystemResource.ChildOf rdf:resource="#ID2"/>
    <cim:ConductingEquipment.Terminals rdf:resource="#ID5"/>
  </cim:Breaker>
  <cim:CurrentRelay rdf:ID="ID4">
    <cim:Naming.name>Relay 1A</cim:Naming.name>
    <cim:ProtectionEquipment.Operates rdf:resource="#ID3"/>
  </cim:CurrentRelay>
  <cim:Terminal rdf:ID="ID5">
    <cim:Naming.name>Terminal101B</cim:Naming.name>
    <cim:Terminal.ConductingEquipment rdf:resource="#ID3"/>
    <cim:Terminal.ConnectivityNode rdf:resource="#ID6"/>
  </cim:Terminal>
  <cim:ConnectivityNode rdf:ID="ID6">
    <cim:Naming.name>Node211B</cim:Naming.name>
    <cim:ConnectivityNode.Terminals rdf:resource="ID5"/>
    <cim:ConnectivityNode.Terminals rdf:resource="ID7"/>
  </cim:ConnectivityNode>
  <cim:Terminal rdf:ID="ID7">
    <cim:Naming.name>Terminal21</cim:Naming.name>
    <cim:Terminal.ConductingEquipment rdf:resource="#ID8"/>
    <cim:Terminal.ConnectivityNode rdf:resource="#ID6"/>
  </cim:Terminal>
  <cim:TransformerWinding rdf:ID="ID8">
    <cim:Naming.name>T1-winding21</cim:Naming.name>
    <cim:ConductingEquipment.Terminals rdf:resource="#ID7"/>
  </cim:TransformerWinding>
</rdf:RDF>

```

Figure 1.7: Document CIM-XML pour le diagramme de la figure 1.3

Quelques éléments de définitions syntaxiques dans le document CIM-XML :

- L'élément racine du document est de type **rdf:RDF**.
- Le nom d'espace de **rdf** doit être déclaré : <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

- Le nom d'espace du modèle CIM de EPRI doit également être déclaré sous la forme : <http://iec.ch/TC57/2000/CIM-schema-<version>#> . La version actuelle est **cim10** (espace de nom = <http://iec.ch/TC57/2000/CIM-schema-cim10#>).
- Chaque élément principal d'un document CIM-XML est une instance d'une classe CIM. Chacun de ces éléments est précédé du préfixe "**cim:**". Un élément CIM est considéré en fait comme une ressource avec un identificateur (ID) unique qui lui est associé (avec **rdf:ID**).
- Chaque ressource CIM peut posséder un ou plusieurs champs et propriétés.
- Les propriétés pour les éléments CIM sont elles-mêmes des ressources CIM et elles peuvent être décrites dans des déclarations subséquentes.

L'outil **CIMValidate**⁸ permet de valider les modèles de réseau CIM. Il valide des fichiers CIM-XML – ou tout document RDF – à partir de son schéma RDF.

1.6 Conclusion

Le modèle d'information commun (CIM) est un modèle abstrait représentant les principaux objets normalement utilisés par une utilité électrique. Il est constitué de paquets composés de classes décrivant ces objets ainsi que leurs relations. Actuellement, le modèle CIM de base existe sous forme de modèle électronique (modèle UML Rose) et sous forme de norme CEI (document Word). Dans le modèle UML Rose, tous les paquets du CIM (normes 61970-3xx : -301, -302, -303) sont représentés. Par contre, pour la normalisation CEI, seule la partie 301 est approuvée. Les autres parties sont en cours.

Le modèle CIM fournit une syntaxe commune pour la description de systèmes électriques. Il est accepté par plusieurs douzaines d'utilités à travers le monde :

- En Amérique du Nord, le modèle est utilisé par les RTOs (Regional Transmission Operators), les ISOs (Independent System Operators) et également par le NERC.

⁸ Plus d'informations sur CIMValidate disponibles sur : <http://www.langdale.com.au/validate/index.html>

- Il est endossé par plusieurs organismes de normalisation
- Plus de quarante fournisseurs vendent des applications / produits basés sur le CIM.

Le modèle CIM constitue ainsi une base non seulement pour l'intégration de modèles mais également pour l'échange d'information entre utilités.

Pour pouvoir permettre l'échange de données, la méthodologie RDF est utilisée afin de créer le document CIM-XML. Ce dernier emprunte le vocabulaire RDF pour coder le modèle abstrait CIM, en utilisant comme langage syntaxique le XML.

Finalement le document CIM-XML est le document de base pour l'échange de données.

Pour valider le modèle d'échange, huit séries de tests d'interopérabilité ont été organisées par le NERC à ce jour. Ces tests regroupaient les vendeurs de logiciels desservant l'industrie des services publics d'électricité et visaient à mesurer la capacité de leurs produits logiciels à échanger et interpréter correctement les données d'un réseau d'électricité basé sur le standard CIM-XML. Les tests réalisés étaient des essais d'importation et d'exportation de modèles complets, partiels et incrémentiels de réseau fournis.

Pour l'importation, il s'agit de convertir la représentation CIM-XML d'un réseau en une représentation interne propre au produit logiciel. Une fois le réseau importé, une application d'écoulement de puissance peut être exécutée pour valider le contenu et la traduction appropriée entre les formats propriétaires et le CIM. Par la suite, le fichier importé peut éventuellement être exporté. L'exportation désigne le processus inverse, sachant que des outils comme le XML/RDF Validator Tool sont utilisés par la suite pour valider le format du document XML exporté et la conformité de XML avec la syntaxe RDF.

Le tableau en annexe A résume le contenu de ces tests, ainsi que les vendeurs impliqués. Après chaque batterie de tests, les problèmes rencontrés sont répertoriés et des propositions sont éventuellement faites pour une représentation CIM-XML plus efficace

des éléments électriques. Les rapports de ces tests sont disponibles sur le site de l'EPRI (www.epri.com).

Le logiciel EMTP-RV n'a jusque-là pas été impliqué dans ces tests d'interopérabilité et nous nous proposons d'étudier la faisabilité pour EMTP-RV d'importer des modèles de réseaux au format CIM-XML. Nous supposons pour cela que les modèles de réseaux qui seront importés sont déjà représentés sous le format CIM-XML.

Des solutions vont être élaborées en guise d'interface entre le modèle CIM-XML et le logiciel EMTP-RV. Pour le test des interfaces réalisées, nous allons d'abord utiliser un petit modèle de réseau à 6 barres dont le schéma est présenté à la figure 1.8.

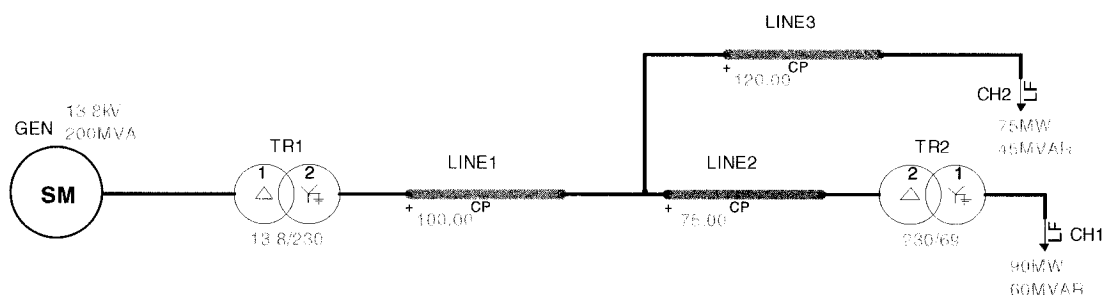


Figure 1.8 : Réseau test à 6 barres

La description de ce réseau ainsi que son document CIM-XML (res_PM.xml), sont présentés en annexe B.

Le test sera également fait sur un modèle de réseau à 60 barres (fichier cim.xml) pris du site intranet de l'IREQ http://pcsatt.ireq.ca/cimnerc/index_fichiers/Exemples.htm. Ce fichier est présenté dans le CD-ROM.

On retrouve dans ce réseau beaucoup d'objets du modèle CIM. Pour juger de la taille de ce réseau, le tableau 1.1 montre le nombre d'instances des classes d'objets CIM impliqués.

Tableau 1.1 : Objets CIM dans le modèle de réseau à 60 barres

Objet CIM	Quantité	Objet CIM	Quantité
ACLineSegment	46	Measurement	44
BaseVoltage	10	MVArCapabilityCurve	16
Breaker	244	PowerTransformer	54
BusbarSection	102	RectifierInverter	4
Company	4	RegulationSchedule	44
Compensator	22	StaticVarCompensator	1
ConnectivityNode	310	SubControlArea	6
ControlAreaOperator	4	Substation	39
CurveSchedData	184	SynchronousMachine	27
DCLineSegment	3	TapChanger	52
FossilFuel	30	Terminal	892
HostControlArea	4	ThermalGeneratingUnit	18
IncrementalHeatRateCurve	18	TransformerWinding	108
Line	52	VoltageLevel	64
LoadArea	8	EnergyConsumer	34

Les sections suivantes vont :

- décrire les procédures pour lire le fichier XML (document CIM-XML) et récupérer les données CIM y figurant ;
- présenter des solutions pour le transfert de ces données vers EMTP-RV.

Nous nous proposons de travailler dans l'environnement .NET de Visual Studio qui, comme nous le verrons dans le chapitre 3, présente plusieurs avantages.

CHAPITRE 2 TRAITEMENT DE DOCUMENTS XML

Afin de pouvoir utiliser les données contenues dans le document CIM-XML, il est nécessaire de traiter le fichier XML. Le consortium W3C définit le « Processeur XML⁹ » ou l'« API XML » comme un module de logiciel capable de lire les documents XML et de fournir un accès à leur contenu et leur structure. De façon plus familière, le terme "parser" (ou analyseur syntaxique) est aussi utilisé.

2.1 Analyse syntaxique

Comme l'illustre l'image ci-dessous, le parser lit le document XML (sa structure) et élabore un arbre en mémoire. Cet arbre construit est une copie/image exacte de l'arbre structural dans le document XML. L'application le manipule comme s'il était le document XML. En réalité, l'application consomme le contenu du fichier XML à travers le parser.

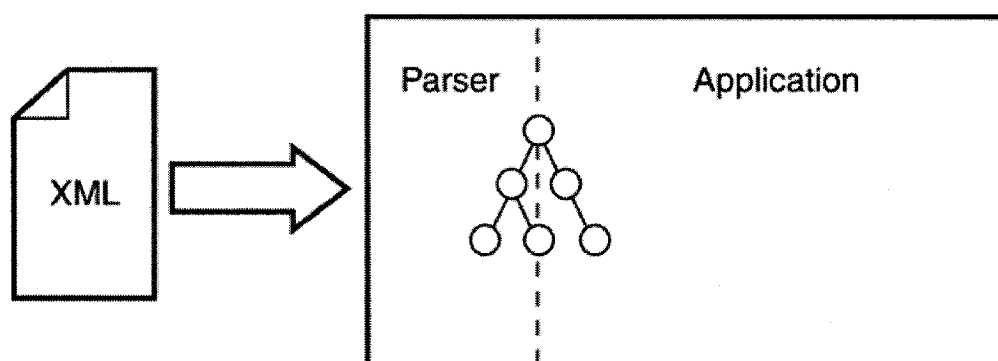


Figure 2.1: Architecture d'une API XML

L'approche utilisée ici pour interfacer un parser avec une application est basée sur des objets. La norme pour cette interface est le Document Object Model (DOM) [9][10] publié par le consortium W3C.

⁹ Ou encore Parser XML – analyseur syntaxique XML

2.1.1 Document Object Model

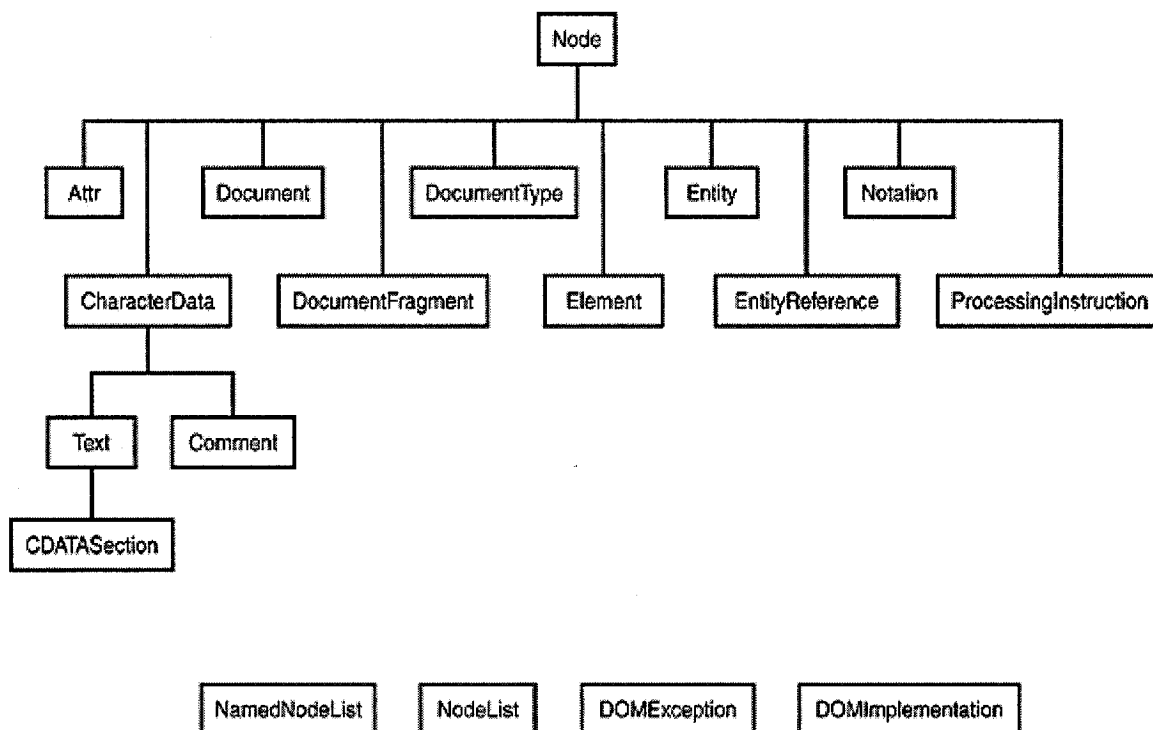
Le DOM est une interface de programmation (API) indépendante de tout langage de programmation et de toute plate-forme qui permet d'accéder à la structure et aux données contenues dans un document XML. Le DOM, non seulement, parcourt le document XML, mais en plus va fabriquer une représentation en mémoire. Cette représentation est un arbre schématisant de façon structurée et orientée objet le contenu d'un document XML.

Pour construire cet arbre logique, il est nécessaire de parcourir et de mémoriser l'intégralité du document avant de pouvoir effectuer les traitements voulus. Pour cette raison, les programmes utilisant DOM ont souvent une empreinte mémoire volumineuse en cours de traitement : le DOM est gourmand en ressources et en temps.

Le DOM définit l'interface programmatique qui devrait être employée pour traverser un arbre XML et pour manipuler les éléments, les valeurs et les attributs : des classes d'objets pour les éléments, les attributs, les expressions littérales, ... sont définies. La figure 2.2 montre les objets les plus importants dans la hiérarchie DOM. Douze types de nœuds différents (Document, Élément, Attribut, Texte, Commentaire, ...) sont définis. Pour plus d'information sur chacun de ces objets, consulter la référence [10].

Le DOM définit diverses méthodes et propriétés permettant notamment d'extraire des données de n'importe quel élément d'un document XML.

De même, le DOM définit les types de relation entre chaque nœud, et des directions de déplacement dans une arborescence XML. Les propriétés *parentNode*, *childNodes*, *firstChild*, *lastChild*, *previousSibling* et *nextSibling* permettent de retourner respectivement le père, les enfants, le premier enfant, le dernier enfant, le frère précédent et le frère suivant du nœud courant.



Source : <http://www.informit.com>

Figure 2.2: Objets dans la hiérarchie DOM

2.1.2 Limites

L'analyse syntaxique (ou *parsing* en anglais) des documents XML présente des limitations. En effet, le *parsing* marche bien lorsqu'on peut accéder et lire distinctement les données des éléments du document XML. Toutefois, manipuler l'information à l'intérieur du document lui-même peut vite s'avérer lourd. C'est le cas des documents CIM-XML. Cela est lié, entre autres, à la représentation "node/breaker" du modèle CIM alors que souvent les outils comme EMTP-RV utilisent une représentation "bus/branch" [18]. Par exemple, avec CIM, chaque terminal d'un élément conducteur est différencié de son nœud de connectivité. Afin de transférer ces données CIM vers EMTP-RV, l'information dans le document CIM-XML doit être manipulée de sorte à relier chaque terminal au nœud de connectivité associé. Ce travail devient vite fastidieux avec le DOM.

Aussi, utilise-t-on la sérialisation XML [12].

2.2 La sérialisation

La sérialisation est un processus de conversion d'un objet en un formulaire (état persistant) qui peut être transporté facilement.

Il y a deux principaux types de sérialisation :

- la sérialisation binaire : elle convertit le type de données en un flux de données binaires
- la sérialisation XML : elle convertit le type de données en un flux de données XML (ASCII), lequel peut par la suite être converti en un document XML.

Un objet est donc sérialisé lorsqu'il est sous un format binaire ou un format XML.

La désérialisation est le processus inverse qui consiste à convertir un flux XML ou binaire en un type d'objet.

Dans notre cas, comme le fichier d'entrée est un document XML, nous allons plus nous intéresser à la sérialisation XML (la désérialisation en fait) par la suite.

La sérialisation XML ne sérialise que les champs publics et les valeurs des propriétés d'un objet dans un flux XML. La sérialisation n'inclut pas d'information de type.

Pour le projet, nous allons utiliser l'environnement .NET pour mettre en oeuvre la sérialisation XML. La plate-forme .NET fournit un ensemble d'outils modernes très puissants au service de l'interopérabilité. Le chapitre 3 fournit plus de détails sur l'architecture de la plate-forme .NET et évoque les fonctionnalités qui seront utilisées pour le traitement des fichiers XML.

CHAPITRE 3 LA PLATE-FORME .NET

.NET (DotNet) est un standard proposé par la société Microsoft, pour le développement d'applications modernes conçues autour des standards de l'internet (XML, SOAP, WSDL, HTTP). Visual Studio est l'environnement de développement professionnel pour la création de ces applications.

La plate-forme .NET a été élaborée en s'appuyant sur une communauté d'utilisateurs et a abouti à l'élaboration de spécifications. Ces spécifications ont été ratifiées par l'ECMA (European Computer Manufacturers Association) qui est un organisme international de standardisation. Cet effort de standardisation a permis l'émergence de plates-formes portées par des entreprises tierces et disponibles sous un grand nombre de systèmes d'exploitation.

Généralement, on parle de "Framework" .NET pour designer l'ensemble constitué des interfaces API offertes et de l'infrastructure d'exécution.

3.1 Architecture

Le framework .NET comprend :

- *Un environnement d'exécution :*

- Un moteur d'exécution, appelé **CLR** (Common Language Runtime), permettant de compiler le code source de l'application en un langage intermédiaire, baptisé MSIL (Microsoft Intermediate Language) et agissant comme une machine virtuelle. Le CLR gère l'exécution du code et procure des services qui simplifient le processus de développement, comme la gestion automatique de la mémoire, la gestion de la sécurité, les autorisations d'exécution. C'est pour cela que le framework .NET est qualifié d'environnement "managé".
- Un environnement d'exécution d'applications et de services Web, appelé **ASP.NET**

- Un environnement d'exécution d'applications lourdes, appelé WinForms.
- **Des services**, sous forme d'un ensemble hiérarchisé de classes appelé *Framework Class Library (FCL)*. FCL est une bibliothèque orientée objet de classes regroupées dans des espaces de noms et définies dans des assemblages (*assembly* en anglais). On y retrouve :
 - la bibliothèque des classes de base (*Base Class Library*) rassemblant les classes permettant de fournir les outils de base aux applications
 - deux bibliothèques de classes d'accès aux données :
 - tout d'abord ADO.NET permettant l'accès sous format XML aux interfaces de bases de données et aux fichiers XML
 - Les classes XML permettant de manipuler les données XML

La figure 3.1 montre les différentes couches du *framework*. A la base, se trouve le CLR. Celui-ci est nécessaire pour exécuter le code intermédiaire. Une fois le code présent dans le CLR, il va enfin pouvoir être compilé par le JIT (Just In Time) Compiler en langage natif de la machine.

Le code qui s'exécute sous le contrôle du runtime (moteur d'exécution) est appelé *code managé*. Inversement, le code qui s'exécute en dehors du runtime est appelé *code non managé*. Les composants COM, les interfaces ActiveX sont des exemples de code non managé.

La couche au dessus du CLR est la BCL (Base Class Library) qui regroupe des classes permettant les manipulations de chaînes de texte, la gestion des entrées/sorties, tâches, des communications réseaux, le design des interfaces réseaux, ...

La couche suivante est composée des deux bibliothèques d'accès aux données : ADO.NET et les classes XML.

La dernière couche, la plus élevée, est utilisée pour la création de Services Web, de pages Web (ASP.NET) et d'applications Windows.

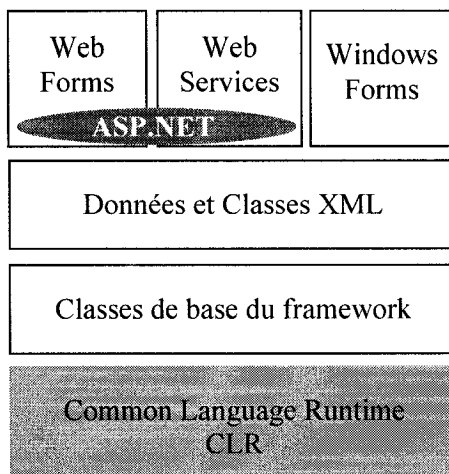


Figure 3.1: Le framework .NET

3.2 Environnement multi-langage

Le framework .NET a été conçu pour permettre l'interopérabilité entre différents langages. Grâce au CLR, la plate-forme .NET est indépendante de tout langage de programmation. En effet, les applications .NET, quel que soit le langage utilisé, sont compilées dans un langage intermédiaire (IL). Par la suite, le code IL est transformé en code machine lors de l'exécution lorsque le système d'exploitation et le processeur sont connus. Une application .NET peut ainsi être déployée sur n'importe quelle plate-forme compatible .NET sans être recompilée.

Plus d'une vingtaine de langages sont supportés par le framework. Parmi les langages de programmation, on retrouve J#, C++, Visual Basic, JScript et C#. Ce dernier a été développé spécifiquement pour la plate-forme .NET. Il s'agit d'un langage orienté objet, dérivé de C++ et Java. Il en reprend les caractéristiques majeures, tout en simplifiant son utilisation.

C'est ce langage que nous allons d'ailleurs utiliser pour la programmation.

3.3 Les espaces de noms

Sous .NET, toute classe est repérée dans un espace de noms (*namespace*). Ce procédé permet d'éviter les conflits de noms entre différentes bibliothèques.

Un espace de noms permet de regrouper des classes dans un domaine. "*System*" constitue la racine des espaces de noms.

La gestion des espaces de noms est particulièrement intuitive puisque chaque niveau de sous-espace est séparé des niveaux supérieurs par un point dans le nom. Ainsi, on retrouve dans le framework de nombreux espaces de noms, chacun étant spécialisé dans un type de service. Par exemple, l'espace de noms *System.Xml* fournit toutes les classes pour le traitement XML. Pour ce qui est de la sérialisation XML, les classes s'y référant sont dans l'espace de noms *System.Xml.Serialization*.

Ce procédé, en plus de prévenir les conflits de noms, permet de hiérarchiser les bibliothèques.

3.4 Assemblage .NET

Dans .NET, un assemblage est une librairie de codes compilés en IL (Intermediate Language). Un assemblage est physiquement contenu dans un fichier PE (portable / exécutable) qui a une extension .exe ou .dll.

Les assemblages sont les équivalents .NET des fichiers .exe ou .dll de Windows. Il s'agit d'un exécutable, accompagné d'un manifeste et d'une description de l'assemblage. Le manifeste d'un assemblage contient les références des autres assemblages qu'il utilise et leurs versions.

Dans les cas des assemblages partagés (*shared assembly*), un mécanisme permet d'avoir plusieurs versions d'un même assemblage. Le but est d'éviter, par exemple, d'avoir des conflits entre les applications suite à des mises à jour de la librairie DLL utilisée.

Pour cela, les assemblages partagés doivent être signés par une paire de clés publique¹⁰ et privée. L'identité¹¹ de l'assemblage ainsi que la clé publique, avec la signature numérique, permettent de doter à l'assemblage d'un "nom fort"¹².

¹⁰ La clé publique permet de signer numériquement l'assemblage.

¹¹ Il s'agit du nom textuel de l'assemblage, de son numéro de version et des informations de culture.

¹² Le nom fort garantit l'unicité du nom en s'appuyant sur la paire de clés uniques. De plus, le nom fort protège l'enregistrement en ligne d'un assemblage car il garantit que nul ne peut produire une version ultérieure de votre assemblage.

Le répertoire GAC (Global Assembly Cache) permet de stocker les assemblages partagés grâce au nom fort qui les identifie de façon unique.

En résumé, pour retrouver un assemblage dans le système, il faut que l'assemblage soit signé et enregistré dans le GAC. L'outil gacutil.exe¹³ (Global Assembly Cache Tool) permet l'enregistrement d'un assemblage dans le GAC. L'utilitaire sn.exe¹⁴ (Strong name Tool) permet de générer de nouvelles clés (pour la signature).

3.5 Composant COM

3.5.1 Description

Le Component Object Model (COM) est un modèle de Microsoft, inventé pour simplifier l'intégration des logiciels. Il s'agit d'une architecture orientée objet spécifiant comment créer des objets ou composants logiciels (comme les DLL) qui peuvent être réutilisés d'une application à une autre, indépendamment du langage de programmation.

Aussi connu sous le nom d'ActiveX, il est utilisé pour permettre le dialogue inter-programmes, ceci dans le but d'échanger des données.

Un composant COM représente du code exécutable contenu dans une bibliothèque de liens dynamiques (.dll) ou un fichier exécutable (.exe). Les composants fournissent un ou plusieurs *objets*, c'est-à-dire des unités de code autonomes qui exécutent des fonctions spécifiques ou tâches. Chaque objet possède des méthodes et des propriétés.

En terme de programmation, le COM présente un ensemble d'interfaces permettant d'utiliser mais aussi d'étendre les fonctionnalités d'un objet COM. Chaque interface COM permet l'accès à un ensemble de méthodes, indépendamment des langages d'appel du composant logiciel et de la machine sur laquelle il est localisé.

Ces interfaces permettent de cacher la logique interne de l'objet COM. Chaque interface doit être unique. Microsoft a donc introduit le GUID (*Global Unique Identifier*) qui est un entier permettant d'identifier de façon unique chaque interface COM.

¹³ Informations sur l'outil gacutil.exe à l'adresse : <http://msdn2.microsoft.com/fr-fr/library/ex0ss12c.aspx>

¹⁴ Informations sur l'outil sn.exe à l'adresse : <http://msdn2.microsoft.com/fr-fr/library/k5b5tt23.aspx>

3.5.2 Serveur et client COM

Le serveur COM est du code dans un langage de programmation donné, qui met à disposition une interface à l'aide d'un objet. L'interface d'un serveur COM est localisable par la librairie COM de la machine d'exécution de l'application, à l'aide d'un identificateur de classe (CLSID). Le CLSID est généré lors de la création du composant logiciel. Il est unique. Il est enregistré dans la base de registre sous Windows.

Le client COM est du code dans un langage de programmation donné, qui met en œuvre un pointeur vers un serveur COM. Un client fait appel à un serveur COM en passant le CLSID à la librairie COM. Cette dernière dispose d'un service SCM (Service Control Manager) chargé de trouver, localement ou sur le réseau, le composant associé au CLSID.

3.5.3 Interopérabilité .NET et COM

En tenant compte que d'autres composants ont été créés dans le passé, il est possible, avec .NET, d'utiliser des composants COM. L'utilitaire `tlbimp.exe`¹⁵ permet de générer un équivalent .NET à partir du fichier .TLB d'un objet COM. Cet utilitaire génère une DLL. Lorsqu'une application .NET fait un appel à la nouvelle DLL générée, l'appel est redirigé vers l'objet COM.

Il est aussi possible d'utiliser un assemblage .NET de la même façon qu'un objet COM. L'outil `tlbexp.exe`¹⁶ prend un assemblage .NET et génère un fichier .TLB. Cela permet à un objet COM d'utiliser un assemblage .NET comme dépendance (ou référence). Cette interopérabilité à double sens est la clé de voute du passage du modèle COM à l'environnement .NET.

¹⁵ Informations sur l'outil `tlbimp.exe` sur <http://msdn2.microsoft.com/fr-fr/library/tt0cf3sx.aspx>

¹⁶ Informations sur l'outil `tlbexp.exe` sur <http://msdn2.microsoft.com/fr-fr/library/hfzzah2c.aspx>

3.6 La sérialisation XML dans .NET

3.6.1 Classes et interfaces

Le framework .NET fournit de nombreuses classes et des interfaces API pour la sérialisation XML. La classe centrale est la classe *System.Xml.Serialization.XmlSerializer*¹⁷ et ses méthodes les plus importantes sont les méthodes *Serialize* et *Deserialize*. Cette classe permet la sérialisation de tous les types définis dans le framework .NET en un document XML et vice versa.

Rappelons que la sérialisation XML ne convertit que les champs publics et les valeurs des propriétés d'un objet dans un flux XML.

Il est possible d'utiliser l'attribut [Serializable] pour indiquer qu'on veut sérialiser un type. Autrement, la classe du type de données peut implémenter l'interface *ISerializable*.

Supposons qu'une application .NET ait défini la classe *Test*, marquée comme sérialisable. L'exemple ci-dessous montre comment utiliser le *XmlSerializer* dans .NET pour convertir une instance de la classe *Test*, appelée *testXml*, en un flux de données (stream) XML.

```
XmlSerializer serializer = new XmlSerializer(typeof(Test));  
serializer.Serialize(fileStream, testXml);
```

La méthode *Serialize* de la classe *XmlSerializer* sérialise le contenu de l'objet *testXml* et génère en sortie un "fileStream". Ce stream pointe vers un fichier qui est le document XML représentant la version sérialisée de l'objet *testXml*.

Pour désérialiser, c'est-à-dire retourner une instance d'un objet .NET à partir d'un document XML stocké sous forme d'un fichier, on peut utiliser la méthode *Deserialize* de la classe *XmlSerializer*. Pour notre exemple, cette méthode va retourner un objet *Test* à partir d'un document XML, passé en entrée comme un "stream".

```
XmlSerializer serializer = new XmlSerializer(typeof(Test));  
Test testXml = (Test) serializer.Deserialize(fileStream);
```

¹⁷ <http://msdn2.microsoft.com/fr-fr/library/system.xml.serialization.xmlserializer.aspx>

Comme cela est montré dans l'exemple précédent, le type d'objet vers lequel la désérialisation va se faire doit être défini. En fait, pour notre étude, il est nécessaire de définir les types d'objets CIM et RDF présents dans un document CIM-XML.

Par lui-même, un fichier XML ne définit pas le type de ses données. Une solution à cette considération consiste à définir des types communs dans un schéma XML utilisé comme référence aux fichiers XML de sérialisation.

De façon générale, le schéma XML constitue le point de départ pour l'échange de données à travers XML. Il suffit que les parties prenantes à l'échange de données se mettent d'accord sur le format d'échange des données pour pouvoir ensuite développer chacun son application.

3.6.2 *Le schéma XML*

Le schéma XML est un langage de description du format d'un document XML. Il permet de définir la structure du document XML. Un fichier de description de structure (*XML Schema Description* en anglais, ou fichier *XSD*¹⁸) est donc lui-même un document XML, avec une structure fixée, comme le montre le document XSD de la figure 3.2, correspondant au schéma XML du document de la figure 1.5.

Le schéma XML supporte les notions de schéma (*schema*), de type complexe (*complexType*), de type simple (*simpleType*), d'élément (*element*) et d'attribut (*attribute*).

Un type complexe peut avoir des attributs et/ou contenir d'autres éléments, alors qu'un type simple ne peut pas se composer d'autres éléments et ne peut pas avoir d'attribut. Un élément peut appartenir à un type complexe ou simple mais un attribut ne peut qu'appartenir à un type simple. L'attribut *fixed* est utilisé dans la déclaration d'attributs et d'éléments pour affecter les valeurs particulières à ces éléments et attributs.

Le document XSD de la figure 3.2 comporte des déclarations d'éléments comme des *complexType*. La définition de type montre que l'élément "RESEAU" peut contenir

¹⁸ Le XML Schema Description est une norme approuvée par le W3C.

deux éléments définis comme "complexType" : "zone" et "Lien". L'élément "zone" lui-même peut contenir les éléments "Prod" et "Charge" ainsi qu'un attribut "nom", tous de type `xs:string`.

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema id="RESEAU" xmlns=""
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="RESEAU">
    <xs:complexType>
      <xs:element name="zone">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Charge" type="xs:string"/>
            <xs:element name="Prod" type="xs:string"/>
          </xs:sequence>
          <xs:attribute name="nom" type="xs:string" />
        </xs:complexType>
      </xs:element>
      <xs:element name="Lien">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Ptransit" type="xs:string"/>
          </xs:sequence>
          <xs:attribute name="nom" type="xs:string" />
        </xs:complexType>
      </xs:element>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figure 3.2: Fichier XSD du document XML de la figure 1.5

Dans le document intitulé "XML Schema Part2 : Datatypes" (cf. référence [11]), le consortium W3C spécifie les types de données simples qui sont autorisés dans un schéma XSD. Pour la plupart de ces types (par exemple, `int`, `decimal`, `double`), il existe un type de données correspondant dans .NET. Cependant certains types de données, comme ceux définis dans les classes CIM ne possèdent pas de correspondants dans .NET.

L'outil xsd.exe¹⁹ (XML Schema Definition Tool) dans .NET permet, dans ce cas, de générer des classes de types à partir d'un schéma XSD. Cet outil a d'autres fonctionnalités, notamment, il permet de :

- Générer un schéma XSD à partir d'un fichier XML.
- Générer un schéma XSD à partir des classes (assemblage .dll).

Le chapitre suivant va décrire pas-à-pas la procédure suivie pour créer les types d'objets CIM et RDF présents dans un document CIM-XML en vue de réaliser la désérialisation des documents CIM-XML et d'en extraire l'information qu'ils contiennent.

¹⁹ Informations sur l'outil xsd.exe sur <http://msdn2.microsoft.com/fr-fr/library/x6c1kb0s.aspx>

CHAPITRE 4 EXTRACTION DES DONNÉES CIM-XML : PROCÉDURE PAS-À-PAS

Ce chapitre présente la première étape de l'étude de faisabilité. Elle consiste à décrire la procédure qui vise à récupérer les données contenues dans un document CIM-XML. Pour ce faire, le *framework* .NET sera utilisé pour exécuter la désérialisation XML.

Dans un premier temps, il s'agit de créer une bibliothèque de classes comportant les types présents dans le document CIM-XML (donc classes CIM et RDF) et de mettre cette bibliothèque dans le cache global (registre de base) pour qu'elle puisse être partagée. Une fois ces types définis et accessibles, la désérialisation du document CIM-XML peut alors être réalisée par une application .NET.

4.1 Création de la bibliothèque de classes

Une bibliothèque de types comporte les informations permettant aux clients COM de déterminer les classes figurant sur un serveur .NET, mais aussi de connaître les méthodes, les propriétés et les événements pris en charge par ces classes.

4.1.1 Problématique

Pour créer la bibliothèque des classes, nous avons besoin aussi bien des classes CIM que des classes RDF représentant les éléments présents dans un document CIM-XML. Pour créer ces classes avec l'outil `xsd.exe`, il nous faut leurs schémas XSD.

Jusque-là, seuls les schémas XSD pour les objets CIM²⁰ (CIM de base) sont créés (site de <http://cimuser.org>) mais le lien avec les éléments RDF n'est pas représenté.

Un autre problème avec certains documents CIM-XML visités, c'est qu'ils ne comportent pas toujours un minimum d'attributs définis pour chaque type d'élément

²⁰ Pour trouver les fichiers XSD des objets CIM, se rendre sur le site <http://cimuser.org/> . Ensuite aller sur le lien CIM puis cliquer sur [61968 XML Schemas](#)

CIM. Par exemple, pour un transformateur (*PowerTransformer*), la puissance apparente (*ratingMVA*) est un attribut qui doit être nécessairement défini.

Autre remarque, dans les tests d'interopérabilité, seules 10% des classes CIM sont utilisées [13].

Pour toutes ces raisons, nous avons choisi de partir de plusieurs documents CIM-XML existants pour générer les fichiers XSD et puis les classes de la bibliothèque. En fait, la démarche consiste à créer les fichiers XSD, puis les classes pour chacun des documents CIM-XML visités. Ensuite, il suffit de commencer avec une bibliothèque de classe et de la compléter avec les classes fournies par les autres fichiers XSD.

4.1.2 Procédure pas à pas

4.1.2.1 Création de schémas XML

Pour chaque document CIM-XML, ses schémas XSD sont créés en utilisant l'utilitaire `xsd.exe`. Par exemple, pour le document CIM-XML de la figure 1.7 nommé "`cim_doc.xml`", il suffit de taper dans la fenêtre de commande :

`xsd.exe cim_doc.xml`

Cette commande génère par la suite autant de fichiers XSD que d'espaces de noms, en l'occurrence deux fichiers XSD pour notre exemple (*cim_doc.xsd*, *cim_doc_app1.xsd*), comme il n'y a que deux espaces de noms²¹ (CIM et RDF). Ces fichiers, ainsi que le document `cim_doc.xml` sont présentés en annexe C. Chaque fichier XSD fait référence aux autres fichiers XSD dont il dépend (avec `xs:import namespace="..."`).

Par ailleurs, l'outil `xsd.exe` n'assume pas que le document source est représentatif de toutes les valeurs possibles (int, double, float, ...) et choisit le type le moins restrictif `xs:string`.

²¹ Les espaces de noms permettent de définir des types uniques et d'identifier différents types de données avec le même nom.

Dans le même sens, cet outil déclare les éléments-enfants d'un élément comme optionnels. N'ayant pas d'information si oui ou non, les éléments-enfants d'un nœud sont requis, il choisit l'option la plus large en fixant l'attribut `minOccurs` à "0" et `maxOccurs` à "unbounded". `minOccurs="0"` signifie qu'aucun élément-enfant n'est requis. `maxOccurs="unbounded"` signifie qu'il est possible d'avoir des multiples de chaque élément-enfant.

4.1.2.2 Création des classes CIM et RDF

L'utilisation la plus répandue de l'outil `xsd.exe` consiste en la génération de classes correspondant aux `complexType`s définis dans le schéma XML. Pour cela, il faut spécifier l'option « /c ».

D'autres options peuvent également être spécifiées, comme :

- le langage de programmation (option « /l »). Le choix existe entre CS (C#, qui est la valeur par défaut), VB (Visual Basic), JS (JScript) ou VJS (Visual J#).
- l'espace de noms du runtime pour les types générés. L'espace de noms par défaut est "Schemas".

En choisissant comme espace de noms *CimEpriTypes* et comme langage de programmation C#, la commande à taper devient la suivante :

`xsd.exe cim_doc.xsd cim_doc_app1.xsd /c /l:CS /n:CimEpriTypes`

Dans la commande, il faut explicitement spécifier tous les fichiers XSD.

Le fichier résultat est un fichier C# (.cs) contenant toutes les classes correspondant soit au `complexType`s des éléments principaux soit à un type référencé par un élément principal.

Remarque : l'outil `xsd.exe` ne produit pas de classe pour un élément principal de type `simpleTypes`.

Le fichier **cimComplet.cs** des classes générées jusque-là est consultable sur le CD-ROM. Ce fichier .cs est la collection de toutes les classes de types générées à partir de plusieurs documents CIM-XML.

Lorsqu'on examine le fichier des classes `cimComplet.cs`, il comporte :

- Une classe `RDF` correspondant à la racine `RDF` du document CIM-XML. Cette classe comporte une liste d'items qui correspondent à tous les objets CIM du document CIM-XML (éléments principaux).
- des classes correspondant aux éléments principaux CIM du fichier CIM-XML comme `Terminal`, `ConnectivityNode`, `Breaker`, `ACLineSegment`, `SynchronousMachine`, ... Chacune de ces classes comporte des attributs et des méthodes définis comme `public`.
- Des classes correspondant aux relations d'association. Lorsqu'un élément a comme attribut une ressource, une classe est alors définie pour l'élément. Par exemple, si dans le fichier XML, on a :

```
<cim:ConductingEquipment.Terminals rdf:resource="#_Borne"/>
```

alors la classe `ConductingEquipmentTerminals` va être créée.

Toutes ces classes sont déclarées publiques et sérialisables.

4.1.2.3 Création de la bibliothèque de classes

Pour produire la bibliothèque de classes dans .NET, nous avons :

1. Créé un assemblage .NET. Pour cela, il faut créer un projet de type "Library Class" avec le fichier *cimComplet.cs* et construire la solution. Cet assemblage est nommé *cim_complet.dll*.
2. Créé une paire de clés et signer l'assemblage. Il suffit pour cela de taper la commande suivante :

```
sn -k cim_complet.snk
```

et d'invoquer le fichier *cim_complet.snk* créé dans les propriétés de la solution.

3. Créé la bibliothèque de types pour l'assemblage. Les bibliothèques de types sont généralement comprises dans des fichiers ayant l'extension .tlb. L'outil

d'inscription des assemblages, regasm.exe²² (Assembly Registration Tool) permet de créer la bibliothèque et de l'inscrire en une seule opération. Il suffit de taper la commande :

regasm /tlb:cim_complet.tlb cim_complet.dll

4. Ajouté l'assemblage au cache global. Il suffit de taper la commande :

gacutil /i cim_complet.dll

Au bout de ces étapes, l'assemblage est installé au niveau du registre de base ; une application cliente peut par la suite importer la bibliothèque et être en mesure d'utiliser les classes CIM définies dans l'espace de noms *CimEpriTypes*.

4.1.2.4 Importation de la bibliothèque

L'utilisation d'une bibliothèque dans une application est une opération à deux étapes.

La première étape consiste à rajouter l'espace de noms des classes de la bibliothèque dans le programme source en utilisant le mot clé "using" pour C# (donc using CimEpriTypes).

La deuxième étape consiste à indiquer au compilateur le chemin des dites bibliothèques. Pour cela, il suffit de rajouter dans les références de l'assemblage, le nom de l'assemblage (fichier .dll) contenant le code des classes de la bibliothèque.

A la suite de ces étapes, une application peut utiliser les types définis dans la bibliothèque de classes.

4.2 Accès aux données CIM-XML

Pour avoir accès aux données dans le document CIM-XML, la désérialisation, décrite dans les sections 2.2 et 3.6, est utilisée. Celle-ci désigne le processus qui consiste à lire un document XML et à construire des objets fortement typés au schéma XML. Les types

²² Informations sur l'outil regasm.exe sur <http://msdn2.microsoft.com/fr-fr/library/tzat5yw6.aspx>

d'objets sont ceux définis dans l'espace de noms `CimEpriTypes` (assemblage `cim_complet.dll`).

Pour que l'application cliente que l'on va créer puisse avoir accès à la bibliothèque créée, il faut qu'elle ajoute, dans ses références, l'assemblage `cim_complet.dll` et qu'elle importe son espace de noms dans le code source avec `"using CimEpriTypes"`. Cette application cliente est elle-même un assemblage .NET.

4.2.1 Désérialisation

Nous allons utiliser la méthode *Deserialize* de la classe `XmlSerializer` de .NET. Nous avons créé une fonction *deser* qui désérialise le document CIM-XML. Voici son code source :

```

1 private Object[] deser(String fileName)
2 {
3     RDF docRDF=new RDF();
4
5     FileStream stream = new FileStream(fileName, FileMode.Open,
6     FileAccess.Read, FileShare.Read);
7
8     XmlSerializer ser = new XmlSerializer(docRDF.GetType());
9
10    docRDF = (RDF)ser.Deserialize(stream);
11    stream.Close();
12
13    Object[] myObjArray = docRDF.Items;
14    return myObjArray;
15 }
```

Figure 4.1: Fonction *deser* réalisant la désérialisation

Cette fonction prend comme paramètre d'entrée *Filename*, le nom du document CIM-XML et retourne un tableau d'objets correspondant aux éléments CIM présents dans le document XML. Le type utilisé pour la désérialisation est `RDF`, correspondant à la racine du document CIM-XML. Le code à la ligne 3 permet de créer une instance *docRDF* de la classe `RDF`.

Il faut également créer un fichier (`FileStream`) appelé *stream* qui va contenir le document CIM-XML.

Ensuite, une instance `ser` de la classe `XmlSerializer` est construite (ligne 8) en lui précisant le type RDF. A partir de là, la méthode `Deserialize` peut être appelée (ligne 10) et les flux de données XML sont convertis en un objet RDF.

Par la suite, il est possible d'avoir accès à tous les éléments principaux du document XML (ligne 13) avec l'attribut *items* de la classe RDF. Le tableau `myObjArray` est en fait une liste brute des objets présents dans le document CIM-XML. Un objet qui est présent plusieurs fois dans le document XML apparaît autant de fois dans `myObjArray`.

4.2.2 Liste des types

Afin de connaître les types d'objets, nous avons produit la fonction *listTypes*, dont voici le code :

```
public String listTypes(String fileName)
{
    Object[] myObjArr = deser(fileName);
    ArrayList nomTypes = new ArrayList();

    String nameTypes = "";
    for (int i = 0; i < myObjArr.Length; i++)
    {
        nom = myObjArr[i].GetType().Name;
        if (!nomTypes.Contains(nom))
        {
            nomTypes.Add(nom);
            nameTypes += nom + "\n";
        }
    }
    return nameTypes;
}
```

Figure 4.2: Fonction `ListTypes` retournant les types d'objets

Cette fonction retourne l'ensemble des types rencontrés (*nameTypes*) sous forme de chaîne de caractères (`String`) à partir du tableau d'objets retourné par la fonction *deser*.

Le principe de la fonction est simple : la fonction parcourt le tableau `myObjArr` ; si un type est déjà présent dans `nomTypes`, la fonction passe à l'objet suivant ; autrement, elle note le nom du type.

A présent, les types d'objets présents dans un document CIM-XML sont connus et les données XML disponibles, enrobées sous des objets (*Object*).

Dans la suite de l'étude, le travail maintenant consiste à exporter ces données vers le logiciel EMTP-RV. Les chapitres suivants décrivent l'architecture du logiciel EMTP-RV et les solutions proposées.

CHAPITRE 5 TRANSFERT DES DONNÉES CIM-XML VERS EMTP-RV

EMTP (ElectroMagnetic Transients Program) est un logiciel qui permet de simuler des régimes transitoires dans des réseaux électriques. Il peut s'agir aussi de transitoires électromagnétiques que de transitoires électromécaniques. Très largement utilisé par les compagnies de transport et de distribution d'électricité, les constructeurs de matériels, les laboratoires de recherche et développement ou encore les sociétés d'ingénierie, ce logiciel fait référence pour la simulation et l'analyse des phénomènes transitoires électromagnétiques.

Le logiciel EMTP-RV [20] est une nouvelle version complètement restructurée et entièrement réécrite du logiciel EMTP.

5.1 Architecture du logiciel EMTP-RV

EMTP-RV possède une interface graphique GUI (Graphic User Interface) appelée EMTPWorks [21]. Cette interface puissante et conviviale constitue l'éditeur schématique de réseaux électriques. Des circuits monophasés et/ou triphasés – simples ou complexes – peuvent être visuellement construits grâce à la bibliothèque de composants.

La figure 5.1 présente un exemple simple de circuit. Tous les éléments du circuit sont issus des bibliothèques de composants prédéfinis. Chaque composant peut avoir zéro, un ou plusieurs pins, connectés à des signaux. Les signaux raccordent les éléments entre eux. Le raccordement peut se faire directement ou alors en utilisant le nom des signaux, comme illustré à la figure 5.2.

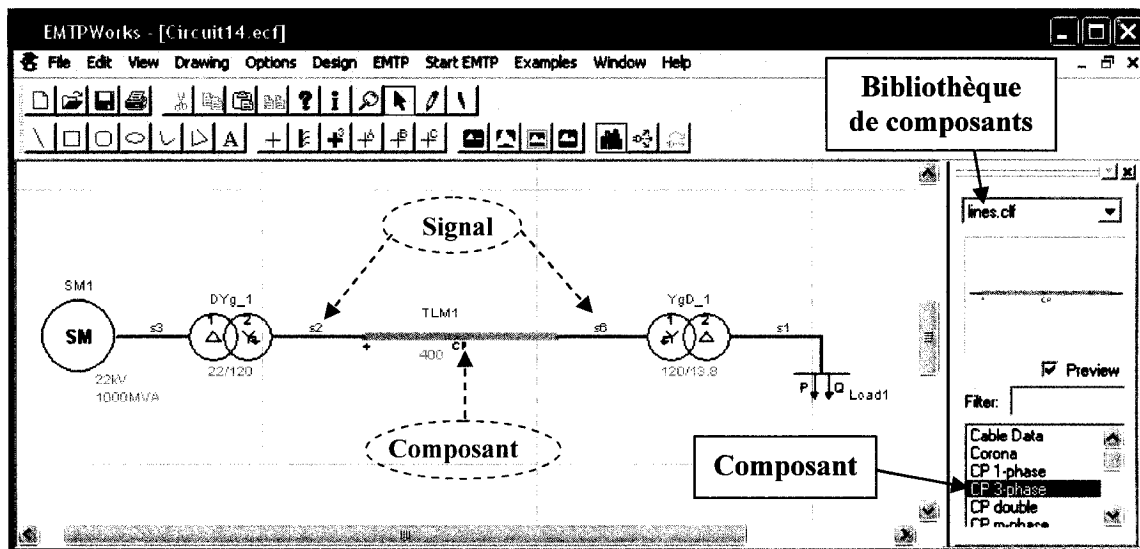


Figure 5.1 : Composant, signal et bibliothèque de composants dans EMTPWorks

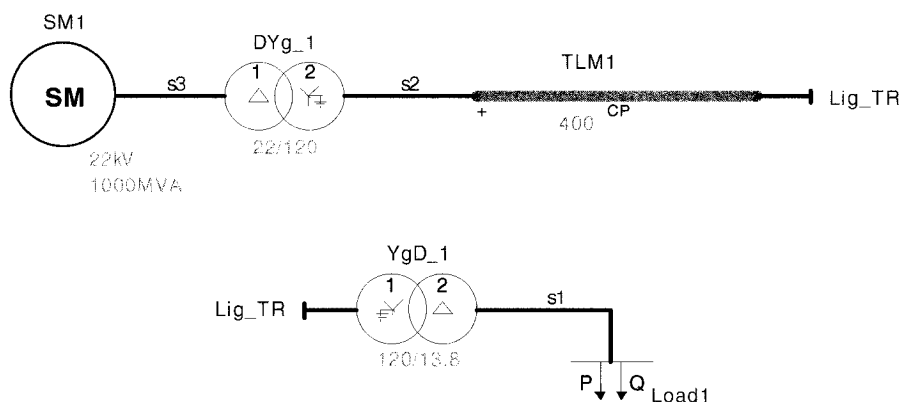


Figure 5.2 : Connexion des éléments par le nom du signal

Dans cet exemple, la ligne TLM1 est reliée au transformateur YgD_1 par le nom du signal (Lig_TR).

Dans EMTPWorks, il est également possible pour un utilisateur de créer ses propres éléments à partir des composants prédéfinis dans les bibliothèques. Par exemple, la figure 5.3 montre un composant représentant une éolienne (composant encerclé en rouge). Le logiciel comporte non seulement un éditeur de symboles (dessin de

l'éolienne) mais également des méthodes automatiques de création de sous-circuits²³ (*Subcircuit*) avec plusieurs niveaux hiérarchiques. Dans l'exemple de la figure 5.3, chacun des éléments du sous-circuit de niveau 1 encadré en pointillé comporte des niveaux de sous-circuits.

Les options de masquage de sous-circuit fournissent une façon commode de définir et d'encapsuler les données d'un sous-circuit dans un script.

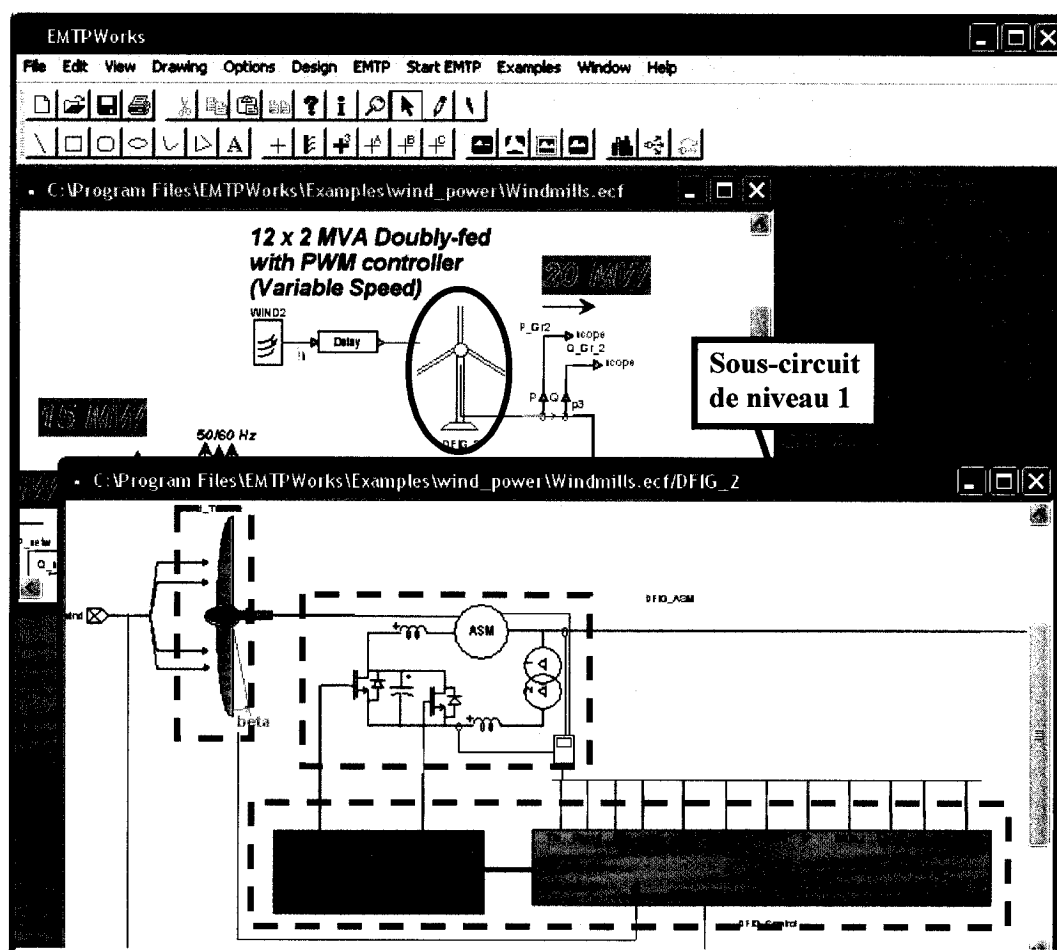


Figure 5.3 : Composant avec plusieurs niveaux de sous-circuits

²³ Un sous-circuit est représenté comme une boîte noire avec ses entrées (input) et sorties (output)

5.1.1 Bibliothèques de composants prédéfinis

Les composants prédéfinis sont répartis en plusieurs bibliothèques « thématiques ». On retrouve dans la première version de EMTP-RV les bibliothèques suivantes, entre autres :

- ***lines.clf*** : pour les modèles de lignes mono- ou polyphasées, les modèles de câbles
- ***machine.clf*** : pour les modèles de machine synchrone, asynchrone, à aimants permanents, ...
- ***transformers.clf*** : pour le modèle de transformateur monophasé idéal, de transformateurs triphasés non idéaux avec le type de connexion des enroulements précisé
- ***RLC branches.clf*** : pour les modèles de résistance (R) et/ou inductance (L) et/ou condensateur (C) simples ou triphasés, ...
- ***options.clf*** : pour désigner les options de simulation à réaliser,
- ***meters.clf*** : pour les sondes (*probe*) de tension, courant et de puissance, des dispositifs de visualisation de signaux (*scope*)
- ***switches.clf*** : pour les modèles de commutateurs (diode, thyristor, interrupteur, ...)
- ***hvdcl.clf*** : pour les modèles de ponts à convertisseurs, les dispositifs d'allumage, ...
- ***control function.clf*** : pour les modèles de fonctions des systèmes de contrôle/commande comme les PID, les PWM, les PLL, ...

Une bibliothèque porte l'extension .clf. Il est possible avec EMTP-RV de créer de nouvelles bibliothèques de composants. Pour plus de détails sur les composants des bibliothèques, se référer à [20].

5.1.2 Programmation orientée objet

Le code de EMTP-RV a été bâti en utilisant la programmation orientée objet. Les composants et les signaux sont définis comme des objets contenant des attributs. Nous ferons d'ailleurs référence par la suite aux composants comme des « objets EMTP-RV ».

Le langage de script est du JavaScript standard avec une extension de EMTPWorks : des méthodes sont rajoutées pour la communication avec EMTPWorks ; ces méthodes permettent d'appeler des méthodes EMTPWorks internes comme par exemple, celles permettant de modifier ou de redessiner les symboles des composants.

Le langage JavaScript est utilisé pour :

- coder les fonctions de données de tous les composants. Les programmes JavaScript (extension .dwj) ont accès à tous les attributs des composants.
- coder les fonctions de bibliothèque

Dans une application JavaScript de EMTP-RV, il est également possible de lancer des appels aux objets ActiveX ou aux programmes créés dans d'autres environnements.

Des scripts peuvent aussi être lancés depuis EMTPWorks.

5.1.3 Simulation des circuits

De EMTPWorks, les éléments d'un circuit à simuler, ainsi que leurs données, sont transmis à EMTP-RV dans un fichier de description appelé « netlist » (*netlist file* en anglais), portant l'extension .net. Ce netlist constitue le fichier d'entrée pour la simulation des circuits. Il contient la description du circuit à simuler, avec des termes et une syntaxe spécifiques. Il contient les données relatives aux composants du circuit, les signaux qui connectent ces composants, les paramètres, ...

Ce netlist peut être généré automatiquement ou sur demande, à partir du fichier (avec l'extension .ecf) créé avec l'éditeur graphique. Mais, il peut également être obtenu de façon manuelle. Par exemple, l'utilisateur peut entrer manuellement la description de son circuit dans un fichier de texte portant l'extension .net, en respectant la syntaxe de définition des éléments.

Dans un premier temps, le netlist sera notre point d'entrée du logiciel EMTP-RV pour transférer les données CIM-XML. Donc nous allons créer de façon manuelle le netlist du système dont les données sont contenues dans le document CIM-XML.

Pour ce faire, il faut, au préalable, établir des correspondances entre les objets CIM et les objets EMTP-RV pour pouvoir par la suite définir les formats netlist.

5.2 Correspondances entre objets CIM et objets EMTP-RV - Formats netlist

Afin de produire le netlist des objets CIM, il faut, au préalable, définir les correspondants de ces éléments dans EMTP-RV. Comme objets CIM, il s'agit par exemple de ACLineSegment, DCLineSegment, Breaker, ...

5.2.1 Remarques générales

1. Dans cette étude, il n'est pas question de chercher des correspondances pour tous les objets CIM de la norme CEI 61970-301. En fait, la recherche de correspondances va se limiter à quelques éléments simples prédéfinis dans les bibliothèques EMTP-RV comme les modèles de lignes triphasées, de lignes à courant continu, les interrupteurs/disjoncteurs, ... Pour des éléments plus complexes, comme les onduleurs/redresseurs, les changeurs de prises dans les transformateurs, ... un travail préliminaire consiste à trouver dans EMTP-RV un modèle valide représenté sous forme de boîte noire (*black box device*).
2. Le modèle CIM fait une distinction entre le terminal d'un élément et son nœud de connectivité. Chaque élément possède un ou plusieurs terminaux (classe *Terminal*). Et chaque terminal est raccordé à un nœud de connectivité (classe *ConnectivityNode*). Pour les éléments CIM dont des correspondances ont été trouvés avec EMTP-RV, nous considérons que le nœud de connectivité dans CIM correspond au signal dans EMTP-RV (cf. figure 5.4). Donc, pour chaque élément, nous allons chercher le nœud de connectivité associé à chaque terminal.

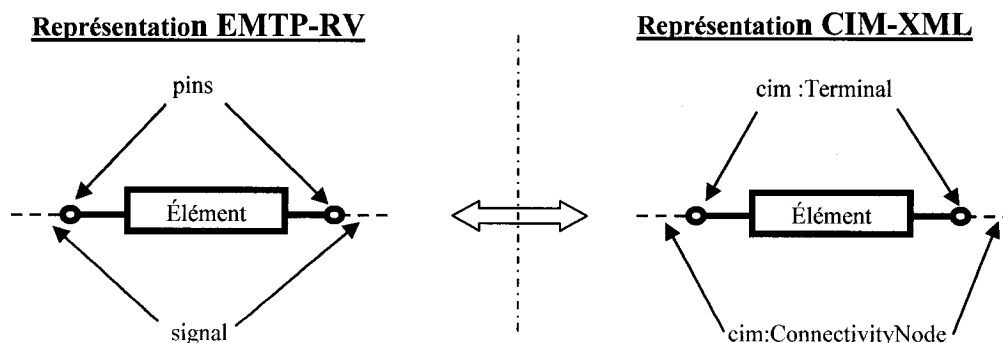


Figure 5.4 : Correspondances dans la représentation d'un élément entre CIM et EMTP-RV

3. Le modèle CIM est orienté "node/breaker". Cela signifie qu'il peut y avoir des « interrupteurs » (*Breaker*) fermés ou ouverts entre des éléments. Par exemple, il est possible de trouver un disjoncteur (*Breaker*) en amont et en aval d'une ligne AC (*ACLineSegment*). Il est intéressant de représenter ces disjoncteurs (leur état ouvert ou fermé) pour connaître la topologie du réseau.
4. Comme, dans EMTP-RV, le nom de chaque élément doit être unique, nous allons considérer la valeur de l'attribut **rdf:ID** défini dans le format d'échange CIM-XML comme le nom de l'équipement. En effet, la valeur de cet identificateur est unique.

5.2.2 Correspondant pour *ACLineSegment*

ACLineSegment désigne une ligne triphasée à courant alternatif (AC). Son correspondant dans EMTP-RV est l'objet "TLM 3-phase" (type `_TLM`). En supposant qu'un bus triphasé est connecté à chaque borne de la ligne, le format netlist est alors le suivant :

```
_TLM;TLM1a;6;2;s1a,s2a
3,0,1,1,1,1,1mH,1pF
_TLM;TLM1b;6;2;s1b,s2b
_TLM;TLM1c;6;2;s1c,s2c
longu r1 11 c1
longu r0 10 c0
0
0
0
```

Les paramètres sont :

- TLM1 est le nom de la ligne AC triphasée

- s1 et s2 sont les signaux triphasés (phases a, b, c) aux bornes de la ligne
- r1 et r0 sont les résistances de séquences positive et homopolaire (en Ω par unité de longueur)
- l1 et l0 sont les inductances de séquences positive et homopolaire (en mH par unité de longueur)
- c1 et c0 sont les condensateurs de séquences positive et homopolaire (en μF par unité de longueur)
- longu est la longueur de la ligne.

Les hypothèses suivantes ont été prises :

- la ligne triphasée est continuellement transposée
- le modèle de distorsion de la ligne n'est pas pris en compte
- le modèle de la ligne est un modèle (r, l, c)

Remarque : si la longueur de la ligne n'est pas fournie dans le document CIM-XML, la longueur longu est fixée à 1 et résistance, inductance et condensateur sont exprimés respectivement en Ω , mH, μF .

5.2.3 Correspondant pour *DCLineSegment*

DCLineSegment désigne une ligne à courant continu. Il est équivalent à une résistance monophasée qui peut être représentée dans EMTP-RV par l'objet "Resistance 1-phase" (type _RLC).

Son format netlist est alors le suivant :

```
_RLC;R1;2;2;s1,s2,
valR,0,0,0,0,
```

où :

- R1 est le nom de la ligne DC
- s1 et s2 sont les signaux monophasés à ses bornes
- valR est la valeur de la résistance de la ligne DC (en Ω).

5.2.4 Correspondant pour Breaker

L'objet CIM "*Breaker*" désigne un appareil mécanique capable d'établir, de porter et d'interrompre des courants sous des conditions normales de fonctionnement et de couper des courants sous des conditions anormales spécifiées.

Comme équivalent dans EMTP-RV, nous avons choisi l'objet "Ideal Switch" (type _Sw0) triphasé. Le netlist correspondant est le suivant :

```
_Sw0;SW1a;2;2;s1a,s2a,
Tclose,Topen,Imargin,0,0,,1,.1,.001,5,
_Sw0;SW1b;2;2;s1b,s2b,
Tclose,Topen,Imargin,0,0,,1,.1,.001,5,
_Sw0;SW1c;2;2;s1c,s2c,
Tclose,Topen,Imargin,0,0,,1,.1,.001,5,
```

Paramètres :

- SW1 est le nom du disjoncteur (*Breaker*)
- s1 et s2 sont les signaux triphasés (phases a, b, c) aux bornes de l'équipement
- Tclose est le temps de fermeture du disjoncteur. Si l'équipement est fermé en régime établi, sa valeur est fixée à -1. S'il est ouvert, Tclose vaut 1E15.
- Topen est le temps d'ouverture du disjoncteur. Si l'équipement est fermé en régime permanent, sa valeur est fixée à 1E15. S'il est ouvert, Topen vaut également 1E15. Par hypothèse, l'équipement *Breaker* est choisi comme un commutateur à l'état ouvert, qui peut se fermer (*closing switch*).
- Imargin est la marge de courant. Nous la fixons nulle.

Les exemples ci-dessous donnent les netlists de disjoncteur fermé et ouvert en régime permanent :

<u>BREAKER FERMÉ</u>	<u>BREAKER OUVERT</u>
_Sw0;SW1a;2;2;s1a,s2a, -1,1E15,0,0,0,,1,.1,.001,5,	_Sw0;SW1a;2;2;s1a,s2a, 1E15,1E15,0,0,0,,1,.1,.001,5,
_Sw0;SW1b;2;2;s1b,s2b, -1,1E15,0,0,0,,1,.1,.001,5,	_Sw0;SW1b;2;2;s1b,s2b, 1E15,1E15,0,0,0,,1,.1,.001,5,
_Sw0;SW1c;2;2;s1c,s2c, -1,1E15,0,0,0,,1,.1,.001,5,	_Sw0;SW1c;2;2;s1c,s2c, 1E15,1E15,0,0,0,,1,.1,.001,5,

Figure 5.5: Netlist pour un Breaker fermé / ouvert en régime établi

5.2.5 Correspondant pour *EnergyConsumer*

L'objet CIM "*EnergyConsumer*" désigne tout utilisateur de l'énergie électrique, autrement dit, tout point de consommation. Cet élément ne possède qu'une seule borne. L'équivalent dans EMTP-RV que nous choisissons pour *EnergyConsumer* est l'objet "PQLoad" triphasé, dont voici le format netlist :

```
_PQload; nom_a;1;1;s1a,
Tens_kVRMSLL,P_MW,0,Q_MVAr,0,60,1,0
_PQload; nom_b;1;1;s1b,
Tens_kVRMSLL,P_MW,0,Q_MVAr,0,60,1,0
_PQload; nom_c;1;1;s1c,
Tens_kVRMSLL,P_MW,0,Q_MVAr,0,60,1,0
```

Paramètres :

- "nom_" est le nom de la charge triphasée
- s1 est le signal triphasé à la borne de la charge (phase a, b, c)
- Tens_kVRMSLL est la tension efficace ligne-ligne exprimée en kV.
- P_MW est la puissance active consommée par phase, exprimée en MW
- Q_MVAr est la puissance réactive consommée par phase, exprimée en MVAr.

Nous avons supposé que la fréquence est à 60Hz.

5.2.6 Correspondant pour *SynchronousMachine*

Le correspondant dans EMTP-RV de l'objet CIM "*SynchronousMachine*" est la machine synchrone triphasée (type : _SM) dont voici le netlist :

```
_SM;nom_a;3;1;s1a,
_SM;nom_b;3;1;s1b,
_SM;nom_c;3;1;s1c,
V=Vbase_a Vbase_b Vbase_c,1kVRMSLL,
P=0 -120 120 1,
f=60 N 0,
R=Vbase ratedMVA Ibase,
S=0 1 1 0 0 1 0 0,
Ra=r x x0
ds=3 1 xd
Rf=xdTrans xdSubtrans 0
qs=3 0 xq,
MS=0 1 1 0,
1 Inertia Damping 0 0 0
0 0 0 0 0
MSu=1 0 0,
PR=100 1e-06 300 0 1e-06,
```


D'autres données comme celles sur les amortisseurs ou encore la saturation ne sont pas précisées dans le modèle CIM. Mais EMTP-RV offre l'option pour le modèle de machine synchrone de ne pas tenir compte de ces éléments.

5.2.7 Correspondant pour *PowerTransformer*

L'objet CIM "*PowerTransformer*" est un appareil électrique qui peut comporter un ou plusieurs enroulements. L'objet CIM *TransformerWinding* permet d'attribuer les enroulements pour un transformateur donné.

Dans EMTP-RV, l'équivalent de *PowerTransformer* est un transformateur triphasé dont la connexion des enroulements (primaire, secondaire) est précisée.

Dans le cadre de ce projet, sept types de connexion pour les enroulements primaire et secondaire sont représentés : YY, YD, YgD, DY, DYg, YgYg et DD.

Hypothèses :

- N'ayant pas l'information de courbe flux/courant dans les objets CIM, la branche de magnétisation est négligée.
- Nous supposons que les transformateurs sont sans changeurs de prises.
- Le type de connexion par défaut est YY.

Le transformateur non idéal, dans EMTP-RV, utilise un sous-circuit comportant 3 transformateurs idéaux ainsi que les résistances et réactances des enroulements, comme le montre le format netlist pour un transformateur de type YY par exemple :

```
<xfmr_YY_unitO_c8188d0a;4;i,j,k,m,
_RLC;RL1;2;2;i,s36,
#R1#,#L1#,0,0,0,
#W1_scope#,
_RLC;RL2;2;2;s31,k,
#R2#,#L2#,0,0,0,
#W2_scope#,
_Tr0;Trs;4;4;s36,j,s31,m
#Ratio#,,,
_x_Lnonl;Lmag;2;2;s36,j,
1,#Phi0#,1e-08,
#Lmag_scope#,
```

```

#ILnonl# #PhiLnonl#
_x_Rmag;Rmag;2;2;s36,j,
#Rm#,,,,,
>
<YY_O_3lee8dcd;8;Ya,Yb,Yc,Youta,Youtb,Youtc,NW1,NW2,
@xmfr_YY_unitO_c8188d0a;xfmr_A;4;Ya,NW1,Youta,NW2,
Phi0=Phiss01
@xmfr_YY_unitO_c8188d0a;xfmr_B;4;Yb,NW1,Youtb,NW2,
Phi0=Phiss02
@xmfr_YY_unitO_c8188d0a;xfmr_C;4;Yc,NW1,Youtc,NW2,
Phi0=Phiss03
>
!
!
@YY_O_3lee8dcd;T1;8;Borne1a,Borne1b,Borne1c,Borne2a,Borne2b,Borne
2c,N1,N2
R1=1.9044;
L1=51.775874;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.4;
Lmag_scope='';
W1_scope='';
W2_scope='';

```

Chaque type de transformateur (suivant le type de connexion) est défini par un nom spécifique. Quel que soit le type de transformateur, les paramètres à fournir sont les mêmes, à savoir :

- R1 et L1 : résistance et réactance de fuite à l'enroulement primaire
- R2 et L2 : résistance et réactance de fuite à l'enroulement secondaire
- Ratio : rapport entre les tensions des enroulements (enroulement 2 vs enroulement 1)
- Borne1 et Borne2 : signaux triphasés connectés aux bornes du transformateur.

5.2.8 Correspondant pour Compensator

L'objet CIM "*Compensator*" permet de représenter des compensateurs série ou shunt. L'attribut *CompensatorType* permet de distinguer ces deux types de compensateurs.

Dans le cas d'un compensateur série, la résistance et la réactance série doivent être fournies. La distinction entre une inductance série ou un condensateur série se fait en se basant sur le signe de la réactance série $x_{\text{série}}$:

- ✓ Si celui-ci est positif, il s'agit d'une inductance série
- ✓ Si le signe est négatif, il s'agit d'un condensateur série

Dans le cas d'un compensateur shunt, il faut se référer au signe de l'attribut `CIM mVarPerSection` (ou *nominalMVar*) pour distinguer une capacitance shunt d'une inductance shunt:

- ✓ Si celui-ci est positif, le compensateur est une capacitance shunt.
- ✓ Si le signe est négatif, il s'agit d'une inductance shunt.

Remarque : Les compensateurs shunts n'ont qu'une borne alors que les compensateurs série en possèdent deux.

Quel que soit le type de compensateur, son équivalent dans EMTP-RV est l'objet 'RLC'.

Le format netlist correspondant pour chacun des cas est donné ci-dessous :

▪ Compensateur série :

```
_RLC;nom_a;2;2;s1a,s2a,
R_serie,L_serie,C_serie,0,0,
_RLC;nom_b;2;2;s1b,s2b,
R_serie,L_serie,C_serie,0,0,
_RLC;nom_c;2;2;s1c,s2c,
R_serie,L_serie,C_serie,0,0,
```

Dans le cas d'une inductance série ($x_{\text{série}} > 0$) , la valeur de $L_{\text{série}}$ est fournie et $C_{\text{série}}=0$.

$$L_{\text{série}} = \frac{x_{\text{série}}}{2\pi f}$$

si le compensateur est un condensateur série ($x_{\text{serie}} < 0$), la valeur de C_{serie} est fournie et $L_{\text{serie}}=0$.

$$C_{\text{serie}} = \frac{1}{2\pi f \cdot x_{\text{serie}}}$$

Paramètres :

- nom du compensateur série ("nom_")
- nom des signaux triphasés connectés aux bornes de l'équipement (s1 et s2)
- valeur de la résistance série (R_{serie})
- valeur de l'inductance série (L_{serie}) ou du condensateur série (C_{serie})

Remarque : résistance, réactance et condensateur sont exprimés respectivement en Ω , H et F.

▪ Compensateur shunt

```
_RLC;nom_a;1;1;s1a,
0,L_shunt,C_shuntuF,0,0,
_RLC;nom_b;1;1;s1b,
0,L_shunt,C_shuntuF,0,0,
_RLC;nom_c;1;1;s1c,
0,L_shunt,C_shuntuF,0,0,
```

La puissance réactive nominale Q_{nom} ainsi que la tension nominale U_{nom} étant fournies, le calcul de L_{shunt} ou C_{shunt} se fait de la façon suivante :

- Si $Q_{\text{nom}} < 0 \rightarrow$ Inductance shunt

$$L_{\text{shunt}} = \frac{U_{\text{nom}}^2}{2\pi f \cdot Q_{\text{nom}}}$$

- Si $Q_{\text{nom}} > 0 \rightarrow$ Condensateur shunt

$$C_{\text{shunt}} = \frac{Q_{\text{nom}}}{2\pi f \cdot U_{\text{nom}}^2}$$

Paramètres :

- ✓ nom du compensateur shunt ("nom_")
- ✓ nom du signal connecté à la borne de l'équipement (s1)
- ✓ valeur de l'inductance shunt (L_shunt) ou du condensateur shunt (C_shunt)

Remarque : résistances, réactances et condensateurs sont exprimés respectivement en Ω , H et μF .

5.3 Résultats

Le tableau 5.2 résume les correspondants EMTP-RV pour les objets CIM considérés, en précisant les hypothèses considérées.

Le netlist, pour un système représenté sous format CIM-XML, est généré en utilisant l'environnement .NET. Le code pour sa génération est présenté dans le fichier nommé *program.cs* qui est rajouté dans le CD-ROM. Ce programme retourne un fichier avec l'extension .net contenant la description des éléments CIM du système.

Le netlist retourné pour le réseau à 6 barres (fichier network_6barres.net) est joint en annexe d. Le tableau suivant montre les quantités d'éléments importés :

Tableau 5.1 : Quantités d'éléments importés pour le réseau à 6 barres

Élément CIM	Description	Quantité
ACLLineSegment	Ligne triphasé à courant alternatif	3
DCLLineSegment	Ligne à courant continu	0
Breaker	Interrupteur / disjoncteur	0
SynchronousMachine	Machine synchrone	1
EnergyConsumer	Charge	2
PowerTransformer (+TransformerWinding)	Transformateur	2
Compensator	Compensateur (shunt ou série)	0

Tableau 5.2 : Correspondances entre des éléments CIM et les objets EMTP-RV

Objet CIM-EPRI	Correspondant Objet EMTP-RV	Commentaires
ACLineSegment	TLM 3-phase	Type: TLM Hypothèses: - ligne AC continuellement transposée - distorsion non prise en compte
DCLineSegment	Résistance 1 phase	Type : RLC
Breaker	Ideal Switch (3 phases)	Type : Sw0 Hypothèses: - Les Breakers sont des "Switch" Maîtres - Le Breaker est du type "Closing Switch"
SynchronousMachine	synchronous machine	Type : SM Hypothèses : - Tensions équilibrées avec un déphasage nul pour la phase a. - Nombre de pôles=10.
EnergyConsumer	Charge (P,Q)	Type : PQload Hypothèse : - Les puissances active et réactive de la charge sont supposées fixes.
PowerTransformer + Transformer Winding	transformers	Type: sous-circuit défini suivant le type de connexion des enroulements. Hypothèses : - 7 types de transformateurs triphasés représentés : YY, YD, DY, DD, YgYg, YgD, DYg. - La branche de magnétisation négligée - Changeurs de prises non pris en compte - Type de connexion des enroulements par défaut : YY
Compensator	RLC Branch	Permet de représenter des compensateurs shunts et série. Type: RLC Hypothèses : - Inductance série si la réactance série a une valeur positive - Condensateur série si la réactance a une valeur négative - Condensateur shunt si la puissance réactive nominale est positive - Inductance shunt si la puissance réactive nominale est négative. - Par défaut, le compensateur est un compensateur shunt.

Pour le réseau à 60 barres (fichier « cim.xml »), l'importation des données est partielle puisque seuls les éléments dont des correspondances ont été établies ont été importés. D'autres éléments comme les onduleurs/redresseurs (*cim:RectifierInverter*) ou encore les changeurs de prises (*cim:TapChanger*) ne sont pas importés dans EMTP-RV (voir Remarque 1, section 5.2.1). Le netlist retourné pour le réseau à 60 barres (network_60barres.net) est présenté en annexe E. Le tableau 5.3 montre la quantité d'éléments importés.

Tableau 5.3 : Quantités d'éléments importés pour le réseau à 60 barres

Élément CIM	Description	Quantité
ACLineSegment	Ligne triphasé à courant alternatif	46
DCLineSegment	Ligne à courant continu	3
Breaker	Interrupteur / disjoncteur	244
SynchronousMachine	Machine synchrone	27
EnergyConsumer	Charge	34
PowerTransformer (+TransformerWinding)	Transformateur	54
Compensator	Compensateur (shunt ou série)	22

Une fois le netlist créé, pour l'importer dans EMTP-RV, il suffit :

1. créer au niveau de l'éditeur graphique EMTPWorks un circuit simple contenant par exemple une résistance
2. générer le netlist de ce circuit
3. remplacer le netlist de ce circuit par le netlist du système CIM-XML à importer.

Une fois cette procédure suivie, le réseau est importé est prêt à être simulé a priori.

5.4 Conclusion

L'adaptation du modèle de représentation CIM au modèle EMTP-RV a requis de faire un certain nombre d'hypothèses. La représentation d'éléments par ces deux types de modèles est différente :

- le modèle EMTP-RV permet une représentation phase par phase, et donc il permet de modéliser des éléments déséquilibrés alors qu'avec CIM, les systèmes sont "balancés".
- Le modèle CIM ne rentre pas dans le détail pour représenter les machines synchrones ; certains paramètres comme le nombre de pôles ou encore l'information sur la saturation ne sont pas représentés.

Le netlist qui est généré comporte toutes les données caractérisant le réseau importé, en respectant la structure netlist prédéfinie dans EMTP-RV. Une fois ce netlist importé dans EMTP-RV, il est possible de faire des simulations dans le domaine temporel.

Comme le netlist suit une structure et une syntaxe bien spécifiques, il est important d'être vigilant et minutieux lorsqu'on veut le modifier ou rajouter d'autres éléments comme par exemple pour la réalisation d'étude de « load flow ». Un écoulement de puissance pourrait en effet valider que les données contenues dans le fichier CIM-XML sont correctement importées dans EMTP-RV.

En raison de la manipulation délicate du netlist et du fait que cette solution ne permet pas la visualisation graphique du circuit CIM importé, nous allons essayer d'explorer d'autres pistes de solutions. Pour ces nouvelles solutions, nous allons privilégier la visualisation graphique dans EMTP-RV du modèle de réseau importé. Une fois importé dans EMTPWorks, il est plus aisé de rajouter des composants ou alors de modifier le réseau importé.

CHAPITRE 6 INTERFACE ACTIVEX

Dans ce chapitre, le travail fait auparavant va être exploitée. Nous allons nous baser sur les correspondances établies dans le chapitre précédent pour créer des méthodes qui pourront être appelées par une application JavaScript dans EMTP-RV.

Ces méthodes doivent pouvoir :

- désérialiser les documents CIM-XML
- retourner des tableaux d'objets représentant les objets CIM (ACLineSegment, DCLineSegment, EnergyConsumer, ...) du fichier CIM-XML.

Ces objets retournés sont en fait des émules des objets EMTP-RV cités dans le tableau 5.1. Ils ont les mêmes noms de champs/attributs que les objets EMTP-RV avec des données pour ces champs correspondant aux données CIM-XML.

Le but est de remplir les champs des objets EMTP-RV avec les données des objets CIM du document CIM-XML. Pour cela, il suffit de "mapper" les objets générés avec les objets EMTP-RV.

Les méthodes appelées par l'application JavaScript vont être exposées par le biais d'une interface ActiveX ou interface COM. Cela implique de créer un serveur COM implémentant les méthodes exposées.

6.1 Serveur COM

Avant de commencer à créer le serveur COM ou classe ActiveX, il est bon de connaître les lacunes de JavaScript (version 5.1). En effet, il est très difficile, voire impossible, d'importer des tableaux d'objets (*Array of Objects*) avec JavaScript. Sachant que la désérialisation des fichiers XML avec .NET retourne des objets, il faut trouver une astuce afin de contourner le problème. Une solution trouvée a été d'« encapsuler » ces objets dans des éléments de type "String" (chaîne de caractères). Autrement dit, les tableaux d'objets représentant les éléments CIM vont être disponibles via le serveur

COM sous forme de String. JavaScript qui dispose d'outils/méthodes capables de traiter les *String* se chargera par la suite de les convertir en Objets.

Les méthodes disponibles via le serveur COM devront pouvoir :

- désérialiser les documents CIM-XML
- retourner sous forme de *String* :
 - un tableau d'objets nommés **TLM** pour les lignes AC (cim:ACLineSegment)
 - des tableaux d'objets nommés **RLC** pour les lignes DC (cim:DCLineSegment) et les compensateurs shunts et/ou série (cim:Compensator)
 - un tableau d'objets nommés **Switch** pour les éléments CIM de type cim:Breaker
 - Retourner un tableau d'objets nommés **Tfo_2wind** pour les transformateurs à deux enroulements (cim:PowerTransformer)
 - Retourner un tableau d'objets nommés **pqload** pour les charges (cim:EnergyConsumer).
 - Retourner un tableau d'objets nommés **SM** pour les machines synchrones (cim:SynchronousMachine).

Avant de donner le contenu des méthodes du serveur COM, nous allons d'abord en expliquer le principe de fonctionnement.

6.1.1 Principe de fonctionnement

Bien que des clients COM puissent appeler le code exposé dans une classe publique par un serveur .NET, le code .NET n'est pas directement accessible aux clients COM. Pour

utiliser le code .NET à partir d'un client COM, il faut créer un proxy²⁴ appelé *wrapper* *CCW* (COM Callable Wrapper), classe enveloppante pouvant être appelée.

Deux conditions sont requises pour la création de cette classe proxy :

1. Tout d'abord, il faut définir une interface et indiquer à la classe qu'elle doit implémenter cette interface. L'extrait de code de la figure 6.1 définit, par exemple, une interface *Icim2empt* et une classe *cim2empt* qui implémente cette interface.
2. Toutes les classes destinées à être vues par les clients COM doivent être déclarées publiques. Il en va de même pour les propriétés et les méthodes (cf. méthode *nombre* dans l'exemple de la figure 6.1).

```
[Guid("5362F574-93FE-44e5-9E76-F2831B549BF2")]
public interface ICim2empt
{
    int nombre(String element);
}

[Guid("3334202C-D207-490c-B8D7-B023A581A909")]
[ProgId("cimServer.cim2empt ")]
public class cim2empt:ICim2empt
{
    //Retourne le nombre de fois qu'un objet CIM apparaît dans le fichier XML
    public int nombre(String fileName, String element)
    {
        Object[] myObjArr = load(fileName);
        int n = 0;
        for (int i = 0; i < myObjArr.Length; i++)
        {
            nom = myObjArr[i].GetType().Name;
            if (nom == element)
            {
                n += 1;
            }
        }
        return n;
    }
}
```

Figure 6.1 : Définition d'une interface et d'une classe implémentant l'interface

²⁴ Un proxy est un élément logiciel qui accepte des commandes émanant d'un composant, les modifie et les transmet à un autre composant

Lorsque l'assemblage contenant l'interface et la classe implémentant cette interface est créé, il reste alors à suivre la procédure consistant à mettre cet assemblage dans le cache (cf. section 4.1.2.3).

Les serveurs COM sont enregistrés et identifiés sur le système grâce à un Global Unique Identifier (GUID). Les interfaces qu'ils exposent sont également enregistrées et identifiées par un mécanisme identique. L'attribut `Guid` permet de faire le lien absolu et unique avec cette interface. La plate-forme .NET fournit un utilitaire *guidgen.exe*, générateur de Guids. Il est également possible de retrouver cette valeur de GUID dans la base de registre.

L'attribut `ProgId` spécifié ("**cimServer.cim2empt**") sera appelé au moment de la création de l'objet `activeX`.

Pour pouvoir utiliser ces attributs, l'espace de noms `System.Runtime.InteropServices` doit être appelé au début (avec C#) par : `"using System.Runtime.InteropServices;"`.

6.1.2 Les méthodes

Le code contenant les membres de la classe *cim2empt* devant être appelés dans JavaScript est présenté dans le fichier **cim2empt.cs**, rajouté au CD-ROM.

Cette classe possède plusieurs méthodes privées et publiques. Mais seules celles qui sont publiques seront exposées.

Parmi les méthodes privées, nous retrouvons la fonction `"load(String fileName)"` qui exécute la désérialisation du fichier XML; la fonction prend en entrée le nom du document CIM-XML et retourne un tableau d'objets. Son code est similaire à celui de la fonction *deser* (cf. figure 4.1).

Les méthodes publiques suivantes appellent la fonction "load(String fileName)" pour désérialiser le document XML. Chacune des méthodes décrites ci-dessous prend en entrée le nom du fichier XML et retourne une chaîne de caractères correspondant au tableau d'objets retournés.

1. La méthode "ACline(String fileName)" retourne un tableau d'objets "TLM" sous forme de String. Dans chaque objet "TLM", on retrouve les données d'un *ACLineSegment* du fichier CIM-XML. L'objet "TLM" a les mêmes attributs que l'objet **CP 3-phase** disponible dans EMTP-RV.
2. La méthode "DCline(String fileName)" retourne un tableau d'objets "RLC_DC" sous forme de String. Chaque objet "RLC_DC" comporte les données d'un *DCLineSegment* du fichier CIM-XML. L'objet "RLC_DC" a les mêmes attributs que l'objet **RLC (monophasé)** disponible dans EMTP-RV.
3. La méthode "Break(String fileName)" retourne un tableau d'objets "Switch" sous forme de String. Chaque objet "Switch" comporte les données d'un *Breaker* du fichier CIM-XML. L'objet "Switch" a les mêmes attributs que l'élément **Ideal Switch** dans EMTP-RV.
4. la méthode "Charge(String fileName)" retourne un tableau d'objets "PQload" sous forme de String. Chaque "PQload" comporte les données d'un *EnergyConsumer* du fichier CIM-XML et a les mêmes attributs que l'élément EMTP-RV du même nom, triphasé.
5. la méthode "Compensateur(String fileName)" retourne un tableau d'objets "RLC_Comp" sous forme de chaîne de caractères. Chaque objet "RLC_Comp" comporte les données d'un *Compensator* du fichier CIM-XML et a les mêmes attributs que l'objet **RLC (triphase)** dans EMTP-RV.
6. la méthode "MachSync(String fileName)" retourne un tableau d'objets "machine_sm" sous forme de chaîne de caractères. Chaque "machine_sm" comporte les données d'un *SynchronousMachine* du fichier CIM-XML et a les mêmes attributs que l'élément **synchronous machine** dans EMTP-RV.

7. la méthode `"transf2wind(String fileName)"` retourne un tableau d'objets `"tfo_2wind"` sous forme de chaîne de caractères. Chaque `"tfo_2wind"` comporte les données d'un *PowerTransformer* à deux enroulements du fichier CIM-XML.

D'autres méthodes publiques ont également été développées ; elles permettent de retourner par exemple la liste des types qu'on retrouve dans un document CIM-XML (`String listTypes(String fileName)`) ou alors le nombre de fois qu'un élément CIM apparaît dans le fichier XML (`int nombre(String fileName, String element)`).

Note : Afin d'éviter les problèmes de traduction de types entre C# et JavaScript, l'attribut `MarshalAs` est utilisé. Celui-ci est placé avant un paramètre d'entrée et une ligne au dessus de la fonction pour spécifier le type du paramètre de retour. Le type `"BStr"` permet de "marshaller" des chaînes de caractères (`String`).

6.2 Client COM JavaScript

6.2.1 Importation des données

Le client COM est une application JavaScript de EMTP-RV. Pour pouvoir utiliser les méthodes du serveur COM, nous créons dans JavaScript un objet `activeX` en spécifiant le *progId* :

```
dom=new activeXObject('cimServer.cim2emtp')
```

A partir de là, en utilisant le document CIM-XML du réseau à 6 barres **res_PM.xml** placé sur le disque C:, les méthodes peuvent être appelées en faisant par exemple :

```
ligne=dom.ACline("C:\\res_PM.xml");  
charge=dom.Charge("C:\\res_PM.xml");  
machine=dom.MachSync("C:\\res_PM.xml");  
transf=dom.Transf2wind("C:\\res_PM.xml");
```

Ce réseau comporte trois lignes AC et donc la méthode ACline par exemple va retourner la chaîne de caractère suivante :

```
TLM[0].name= 'LINE1';TLM[0].nbphases = '3';TLM[0].ctransposed = 'ON';TLM[0].distortioncheck =
'OFF';TLM[0].lccheck = 'ON';TLM[0].zvcheck = 'OFF';TLM[0].ztcheck = 'OFF';TLM[0].lc_runits=
";TLM[0].lc_cunits= 'uF';TLM[0].lc_lunits= 'mH';TLM[0].linelen= '100';TLM[0].propag_grid=
'100\t0.316764\t3.22209182289542\t0.00786994667765574\n100\t0.0243404\t0.923800012715551\t0.012
6000315863469';TLM[0].pin1= 'CN2';TLM[0].pin2= 'CN3';
TLM[1].name= 'LINE2';TLM[1].nbphases = '3';TLM[1].ctransposed = 'ON';TLM[1].distortioncheck =
'OFF';TLM[1].lccheck = 'ON';TLM[1].zvcheck = 'OFF';TLM[1].ztcheck = 'OFF';TLM[1].lc_runits=
";TLM[1].lc_cunits= 'uF';TLM[1].lc_lunits= 'mH';TLM[1].linelen= '75';TLM[1].propag_grid=
'75\t0.316764\t3.22209182289542\t0.00786994667765574\n75\t0.0243404\t0.923800012715551\t0.01260
00315863469';TLM[1].pin1= 'CN3';TLM[1].pin2= 'CN4';
TLM[2].name= 'LINE3';TLM[2].nbphases = '3';TLM[2].ctransposed = 'ON';TLM[2].distortioncheck =
'OFF';TLM[2].lccheck = 'ON';TLM[2].zvcheck = 'OFF';TLM[2].ztcheck = 'OFF';TLM[2].lc_runits=
";TLM[2].lc_cunits= 'uF';TLM[2].lc_lunits= 'mH';TLM[2].linelen= '120';TLM[2].propag_grid=
'120\t0.316764\t3.22209182289542\t0.00786994667765574\n120\t0.0243404\t0.923800012715551\t0.012
6000315863469';TLM[2].pin1= 'CN3';TLM[2].pin2= 'CN6';
```

La chaîne de caractères met en évidence un tableau d'objets TLM avec trois éléments : TLM[0], TLM[1] et TLM[2]. Pour chacun de ces objets, la valeur des attributs est donnée.

De même, pour les deux charges, le résultat retourné avec **dom.Charge("res_PM.xml")** est un tableau de deux objets PQload avec les valeurs de leurs attributs :

```
PQload[0].name= 'CH1';PQload[0].Np= '0';PQload[0].Nq= '0';PQload[0].Ppower= '30';PQload[0].Punits=
'MW';PQload[0].Qunits= 'MVar';PQload[0].Qpower = '20';PQload[0].Voltage = '69';PQload[0].Vunits =
'kVRMSLL';PQload[0].Frequency = '60';PQload[0].RetrieveLFdata = '1';PQload[0].UseRLmodel =
'0';PQload[0].pin1= 'CN5';
PQload[1].name= 'CH2';PQload[1].Np= '0';PQload[1].Nq= '0';PQload[1].Ppower= '25';PQload[1].Punits=
'MW';PQload[1].Qunits= 'MVar';PQload[1].Qpower = '15';PQload[1].Voltage = '230';PQload[1].Vunits =
'kVRMSLL';PQload[1].Frequency = '60';PQload[1].RetrieveLFdata = '1';PQload[1].UseRLmodel =
'0';PQload[1].pin1= 'CN6';
```

6.2.2 Traitement des données

Maintenant que les données CIM peuvent être importées dans une application JavaScript de EMTP-RV sous forme de tableaux d'objets, l'objectif est de charger dans EMTPWorks ces objets pour les rendre visuels.

Pour ce faire, une nouvelle bibliothèque, appelée **CIM.clf** est créée dans EMTP-RV. Celle-ci contient des éléments EMTP-RV correspondant à certains objets CIM, comme le montre le tableau ci-dessous. La nécessité de créer cette bibliothèque CIM.clf repose

sur le fait que, par défaut, certains objets EMTP-RV comme les "Breakers" sont monophasés. Dans la librairie CIM.clf, ces objets apparaissent triphasés.

Tableau 6.1 : Éléments de la bibliothèque CIM.clf

Objet CIM	Bibliothèque CIM.clf	
	Élément EMTP-RV	Caractéristique
ACLineSegment	'CP 3-phase'	Bus triphasé
Breaker	'Ideal Switch'	Bus triphasé
Compensator	'RLC'	Bus triphasé
EnergyConsumer	'PQload'	Bus triphasé

La procédure d'importation des données dans EMTP-RV commence par créer une page blanche pour la représentation du circuit et d'en définir la bordure, l'espacement entre deux éléments ainsi que le point initial à partir duquel placer les éléments :

```
//Aller au nouveau circuit
cct = currentCircuit();

//définir les bordures de la page
pages = cct.pageList();
border = pages[0].border();

//définir le nombre max d'elements à inserer sur une meme ligne
var maxnum, posX, posY;
maxnum = Math.floor((2*border.right)/1820); //Elements espacés de 1.82 pouce
posX_init = border.left + 1200;
posY_init = border.top + 700 ;
posX = posX_init;
posY = posY_init;

cim=load_cim();
```

La fonction **load_CIM** crée une instance, à travers la variable *cim*, de l'ActiveXObject comme défini plus haut.

Le code intégral pour la création et la représentation visuelle des objets EMTP-RV correspondant aux objets CIM est présenté dans le fichier **test_cim.dwj** qui est fourni dans le CD-ROM.

Pour en expliquer le contenu, nous allons juste détailler la fonction ***load_PQload***, qui permet de créer et de représenter dans le circuit défini toutes les charges contenues dans un document CIM-XML. Le code de cette fonction est présenté à la figure 6.2.

La fonction ***load_PQload*** prend comme paramètres d'entrée la variable *cim* et le nom du fichier XML *filename*.

Tout d'abord, il s'agit de connaître le nombre de charges présentes dans le document CIM-XML (ligne 4). Ensuite, le type de charge "PQload" est cherché dans la bibliothèque de composants dans laquelle il se trouve, à savoir CIM.clf (lignes 6 à 8).

Par la suite, on va charger les méthodes et les attributs de l'objet PQload (ligne 10).

Les lignes 15 à 23 permettent de créer et de positionner les éléments de charge sur la page du circuit. Jusque-là, les champs des objets créés ne sont pas encore remplis avec les données CIM. A la ligne 25, les données de charge CIM sont extraites du fichier XML sous forme de chaîne de caractères (Cf. section 6.2.1). Les noms sur ces chaînes de caractères ne sont pas anodins comme ils correspondent mot pour mot aux attributs de l'objet PQload prédéfini dans EMTP-RV.

Ainsi, avec la fonction *eval* (ligne 26), la chaîne de caractère devient un véritable tableau d'objets PQload, avec les champs de ces objets remplis de données CIM.

Les lignes 28 à 32 permettent d'enregistrer les données CIM et également d'affecter et de rendre visibles le nom de ces objets et de leurs signaux (*fonction set_name_pins()*). En fait, la connexion des divers éléments se fait par le nom du signal à leur borne.

On procède ainsi pour tous les éléments présents dans le document XML.

```

1  function load_PQload(cim,filename){
2      try{
3          //nom = "EnergyConsumer";
4          var nb_PQload = cim.nombre(filename,"EnergyConsumer");
5          //get the library
6          var myLib = DWLibrary('CIM.clf');
7          // get the RLC type component
8          var myType = myLib.loadType('PQ load');
9          //Parse all methods for the PQ load device
10         parseScriptFile('pqload_m.dwj');
11         var i,j,k;
12         j=k=0;
13         if(nb_PQload> 0){
14             var PQload = new Array();
15             for(i=0;i< nb_PQload; i++){
16                 if (i%maxnum == 0) {j=0; k++}
17                 //add the device in the circuit at PosX and PosY
18                 posX = posX_init + j*1820;
19                 posY = (posY_init + (i%maxnum)*140)+(k-1)*1120;
20                 var mydev = cct.addDevice(myType,posX,posY);  j++;
21                 /*Create the object
22                 PQload[i] = new oDevice_pqload(mydev);
23             }
24             // get the data from the cim_file
25             var PQload_text = cim.Charge(filename);
26             eval(PQload_text);
27             //Save back the data and set the signal names
28             for(i=0;i< nb_PQload; i++){
29                 PQload[i].SaveData();
30                 set_name_pins(PQload[i].dev,PQload[i].name,PQload[i].pin1,PQ
31                 load[i].pin2)
32             }
33         }
34         posY_init = posY+140;
35     }
36     catch(e){
37         alert('PQload' + e.description);
38     }
39 }
40
41 function set_name_pins(dev,name,pin1,pin2){
42     var dev;
43     dev.setAttribute('Name',name,0); //set the name
44     pins = dev.pins;
45     signal1 = pins[0].signal;
46     signal1.setAttribute('Name',pin1,1);
47     if(pins[1]){
48         signal2 = pins[1].signal;
49         if(pin2 && pin2 !=''){
50             signal2.setAttribute('Name',pin2,1);
51         }else{
52             signal2.setAttribute('Name','GND',1);
53         }
54     }
55 }

```

Figure 6.2 : Code pour l'importation des charges dans EMTP-RV

6.2.3 Résultats

En appliquant le code développé sur le fichier XML test à 6 barres (res_PM.xml), tous les éléments du modèle de réseau sont chargés et peuvent être visualisés graphiquement (cf. figure 6.3).

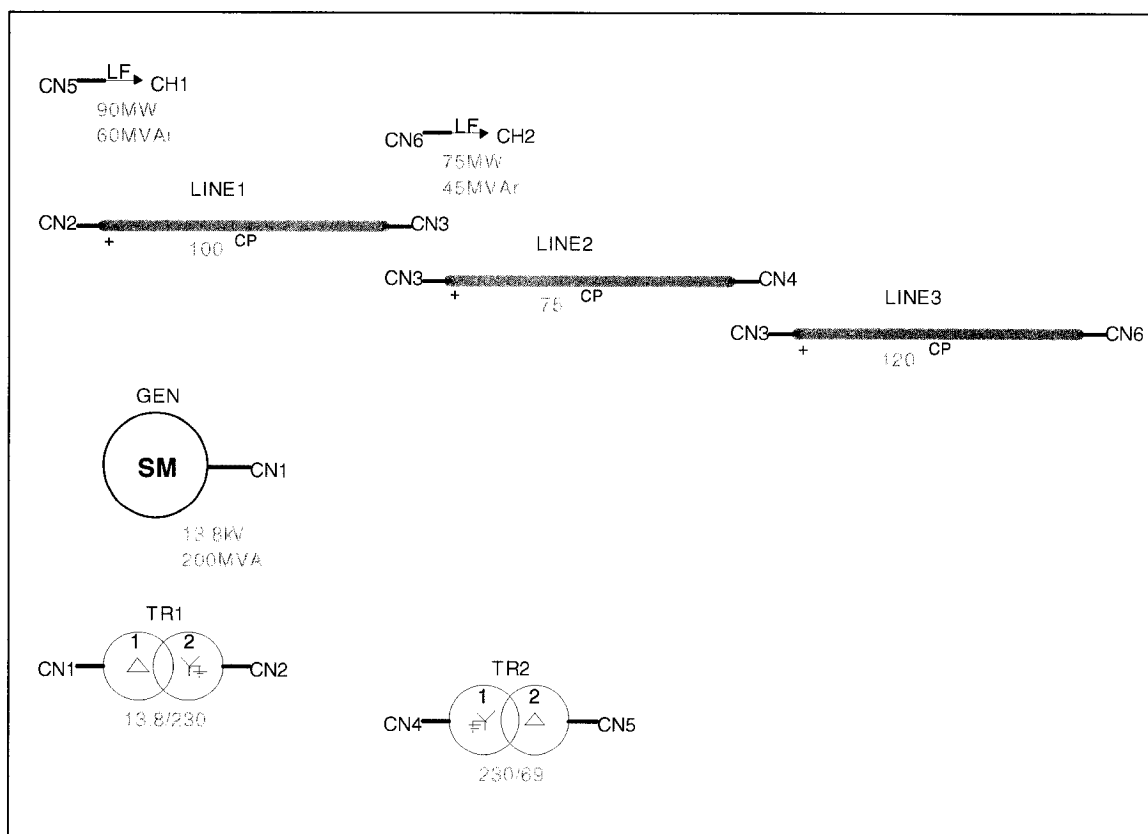


Figure 6.3 : Résultat de l'importation du réseau res_PM.xml dans EMTP-RV

Réarrangés, on peut bien se rendre compte que les éléments sont connectés par nom (cf. figure 6.4). La configuration du circuit est identique à celle du circuit original (figure 1.8).

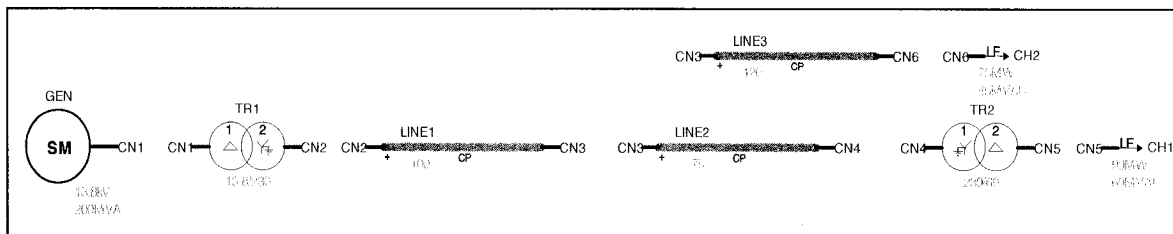


Figure 6.4 : Circuit de la figure 6.3 arrangé

La vérification des données contenues dans ces éléments montre que notre interface fait une bonne interprétation des données CIM. En effet, les résultats d'un écoulement de puissance effectué sur le circuit initial (cf. annexe B) et sur le circuit importé (cf. annexe F) sont quasiment similaires.

Les éléments CIM du fichier cim.xml (réseau à 60 barres) dont les correspondances ont été trouvées avec EMTP-RV (cf. tableau 5.3) ont également pu être importés avec succès. Le circuit importé est présenté en annexe G. Le temps total requis pour importer les éléments est de l'ordre de 3 à 5 minutes. Une vérification rapide a pu montrer que certaines données pour ce réseau sont manquantes. C'est le cas par exemple des données de puissance apparente pour certains transformateurs ou encore de la tension nominale de certaines machines synchrones. Aussi, on peut se rendre compte que trois machines synchrones ne sont pas connectés (pas de signal à leur borne).

6.3 Conclusion

Dans ce chapitre, la démonstration de l'importation correcte par EMTP-RV de données de réseaux sous format CIM-XML a été faite en utilisant une technologie mature et éprouvée qu'est l'interface ActiveX. Un modèle de réseau à 6 barres et un modèle partiel de réseau à 60 barres ont été reproduits graphiquement dans EMTP-RV. Tous les éléments du circuit importé sont reliés par le nom.

Ce qu'il faudrait par la suite, c'est étendre la recherche de correspondances à tous les éléments du paquet « Wires » du modèle de base CIM. On retrouve dans ce paquet tous les éléments généralement présents dans un réseau électrique. Pour certains éléments de

ce paquet comme les redresseurs/onduleurs, les changeurs de prises, qui peuvent être représentés dans EMTP-RV par des boîtes noires, un travail préalable requiert de créer et de valider les modèles de ces éléments dans EMTP-RV.

Juste rappeler que le modèle CIM est statique. C'est-à-dire que le modèle de réseau qui est importé correspond à une certaine topologie. Lorsque la topologie du réseau change (par exemple, un disjoncteur qui devient ouvert), il est possible de faire la modification directement dans le circuit précédemment importé dans EMTP-RV (pour de petites modifications) ou alors importer à nouveau le réseau dans sa nouvelle topologie.

Une limitation de la solution présentée dans ce chapitre, c'est qu'elle basée sur une interface ActiveX (objet COM) créée avec .NET. Le problème, c'est que cet objet COM ne peut être utilisable par un utilisateur ne disposant pas de l'environnement .NET. Les objets COM créés dans cet environnement ne sont pas comme des objets activeX classiques, dans le sens où ils ne peuvent être enregistrés dans le registre de base d'un ordinateur avec la commande classique *regsvr32*. Pour les enregistrer dans le registre de base, il est impératif de travailler dans .NET et de suivre la procédure mentionnée à la section 4.1.2.3.

Dans la continuité du travail fait jusque-là, nous allons explorer la technologie des services Web qui permet la communication à distance, à travers Internet, entre des applications. Cette solution permet de s'affranchir de la limitation de la solution présentée dans ce chapitre, à savoir la nécessité que le *framework* .NET soit installé.

Les services Web font l'objet du prochain chapitre.

CHAPITRE 7 LES SERVICES WEB

Dans ce chapitre, nous proposons une solution pour importer les données CIM-XML vers EMTP-RV basée sur la technologie des services Web. Dans un premier temps, nous allons décrire cette technologie avant de l'exploiter dans le cadre de notre étude.

7.1 Description et concepts

Un service Web (*Web service* en anglais) est une technologie permettant à des applications de dialoguer à distance via Internet, et ceci indépendamment des plateformes et des langages sur lesquels elles reposent. Les services Web permettent aux développeurs d'exposer au monde extérieur des services via un serveur Web. Le but est de permettre à une application de trouver automatiquement sur Internet le service dont elle a besoin et d'échanger des données avec lui.

De façon pratique, un service Web est un composant programmé dans n'importe quel langage et enveloppé dans une couche de normes – la plupart dérivées de XML. Grâce à cette enveloppe, le composant peut dialoguer avec d'autres applications qui, elles-mêmes se conforment aux standards des Services Web.

Les Services Web comprennent :

- un **encodage**. Les données échangées entre les applications sont au format XML.
- un **transport**. Les données échangées peuvent être transportées entre les applications en utilisant des protocoles communs tels que HTTP, FTP, SMTP, et XMPP. Le plus souvent, HTTP est utilisé.
- une **organisation de requêtes/réponses** : les services Web s'appuient sur un ensemble de protocoles d'échanges standardisés, basés sur XML, comme SOAP, XML-RPC, ... Les Web Services sont généralement utilisés en mode RPC (*Remote Procedure Call*). Tout est ramené à des appels de sous-programmes, avec des paramètres entrants et un résultat.

Les services Web présentent l'avantage d'être attaquables depuis n'importe quel langage de développement et de passer facilement à travers les pare-feux²⁵ (*firewall* en anglais).

7.1.1 Architecture

L'architecture des Services Web fait intervenir trois catégories d'acteurs :

- Le serveur Web, fournisseur de services Web. Ce dernier doit développer et publier un ensemble de fonctionnalités (ou méthodes) et les rendre accessibles sur Internet (en accès libre ou privatif via un Extranet) ou sur un réseau local (Intranet).
- Les clients, consommateurs de services Web. Le client Web fait appel au sein d'une application à des fonctionnalités extérieures (méthodes exposées par un serveur Web).
- Les annuaires qui offrent aux fournisseurs la capacité de publier leurs services et aux clients le moyen de localiser leurs besoins en termes de services.

La dynamique de l'architecture se décompose ainsi :

1. la publication du service par le fournisseur dans un annuaire. Le service Web doit alors être déployé et enregistré dans cet annuaire
2. la localisation du service par le client en recherchant dans l'annuaire. Ce dernier communique l'information sur le serveur hébergeant le service
3. et enfin, l'interaction entre le client et le serveur Web.

Le client qui souhaite accéder aux fonctionnalités d'un service Web doit être capable d'envoyer une requête HTTP vers le serveur Web.

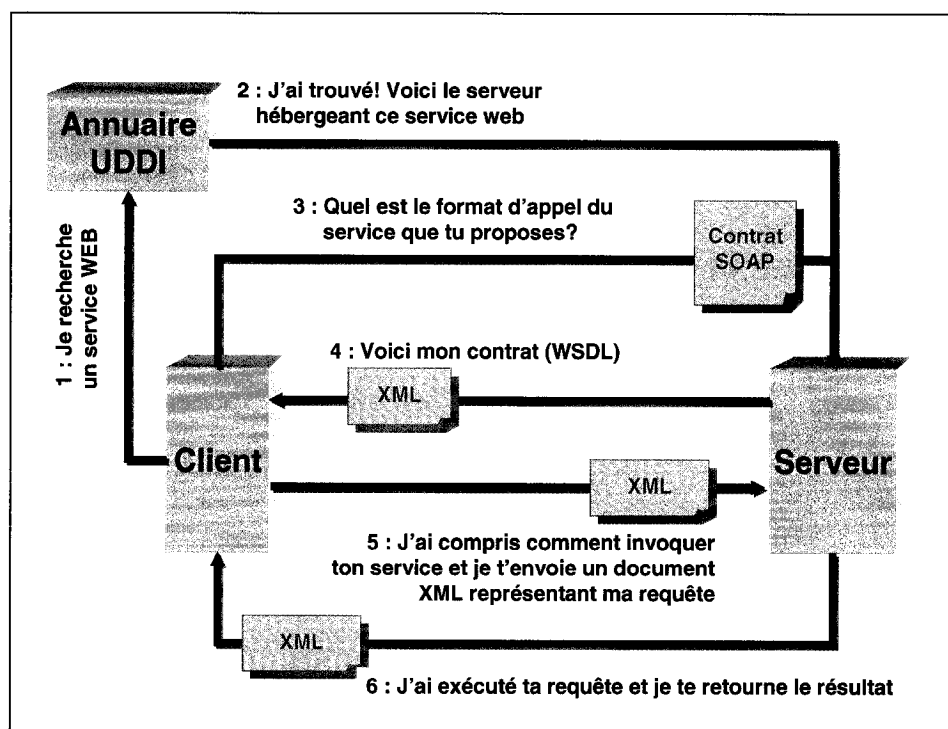
Les messages seront échangés avec le protocole SOAP basé sur XML, protocole qui permet une invocation à distance des services Web (cf. section 7.1.2 pour SOAP).

Le client est capable de connaître la liste, la syntaxe et la documentation des méthodes publiées sur le serveur Web en analysant un fichier fourni par celui-ci, également basé

²⁵ Un pare-feu permet de protéger les ordinateurs, PC, des intrusions extérieures.

sur XML, nommé WSDL (cf. section 7.1.3). Ce fichier WSDL est généralement publié dans un annuaire appelé UDDI (cf. annexe H), par lequel le client peut localiser des services Web.

Le graphe de la figure 7.1 schématise la procédure d'utilisation des services Web par un client.



Source : [15]

Figure 7.1 : Dynamique de l'architecture d'un service Web

Le cycle de vie d'utilisation d'un service Web comporte plusieurs étapes :

1. Tout d'abord, le **déploiement** du service Web, qui dépend de la plate-forme. Le Framework .NET par exemple permet de créer des services Web. Le déploiement aboutit à la disponibilité sur le serveur Web d'un document WSDL décrivant le service.

2. Ensuite, il s'agit d'**enregistrer** le service Web – c'est-à-dire son document descriptif WSDL – dans l'annuaire appelé UDDI.
3. Un client peut par la suite **découvrir** ce service Web en le recherchant dans l'annuaire UDDI. Ce dernier lui indique le serveur hébergeant ce service.
4. Enfin, le client peut **invoquer** ce service en communiquant avec le serveur Web via le protocole SOAP.

7.1.2 Le protocole SOAP

SOAP (Simple Object Access Protocol) est un protocole simple pour l'échange d'informations typées et structurées sur le Web. Reposant sur le langage de balisage XML, SOAP définit la structure des messages échangés par les applications via Internet. Cette structure est dépourvue de toute sémantique de transport ou d'application. SOAP a été construit pour pouvoir aisément être porté sur toutes les plates-formes et les technologies. Il s'agit ainsi d'un protocole modulaire et hautement extensible.

La spécification du protocole SOAP [16] est composée de trois parties :

1. une enveloppe extensible pour l'encapsulation des données. Cette enveloppe SOAP décrit la structure des messages SOAP et la manière dont ils doivent être traités. Cette enveloppe constitue l'unité de base de l'échange entre des processeurs de messages SOAP et constitue la seule partie obligatoire de la spécification. L'espace de noms identifiant l'enveloppe SOAP est donné par :
<http://schema.xmlsoap.org/soap/enveloppe/>
2. Un jeu de règles d'encodage des données, facultatif. Ces règles définissent le mécanisme de sérialisation permettant de construire le message pour chacun des types de données pouvant être échangés. L'espace de noms identifiant la sérialisation SOAP est donné par : <http://schema.xmlsoap.org/soap/encoding/>

3. un modèle d'échange de messages de type RPC (requête/réponse) qui définit une convention pour représenter les procédures d'appels à distance. Cette partie est également facultative.

SOAP peut être associé à n'importe quel mécanisme ou protocole de transport capable de transporter l'enveloppe SOAP, comme HTTP, SMTP, FTP, ... Nous allons pour la suite considérer la liaison entre SOAP et HTTP, comme ce dernier est généralement utilisé.

7.1.2.1 Modèle d'échange de messages SOAP

SOAP permet une communication par message d'un expéditeur vers un récepteur. Les messages SOAP sont des transmissions fondamentalement à sens unique.

Le modèle d'échange est de type RPC :

- Une requête est émise du client vers le serveur Web. Elle contient un seul message (appel sérialisé d'une méthode sur un objet). Une requête se présente sous la forme d'un document XML et contient les paramètres IN et INOUT.
- Une réponse provenant d'un serveur Web est envoyée vers le client. Elle contient également un seul message (retour sérialisé d'un appel de la méthode sur un objet). Une réponse contient les paramètres INOUT et OUT.

Lorsqu'une transmission de message commence (invocation d'un service Web), un message SOAP – ou document XML – est généré. Ce message, envoyé à partir d'un *SOAP Sender*, peut passer par des intermédiaires – nœuds intermédiaires – avant d'arriver au SOAP Receiver. Un intermédiaire SOAP est une entité capable de recevoir et de transmettre des messages SOAP. Ces intermédiaires, sur le chemin d'un message SOAP, traitent des parties du message. Les nœuds intermédiaires ainsi que le SOAP Receiver sont identifiés par un URI.

La figure 7.2 montre un exemple de message transféré entre un client et un serveur.

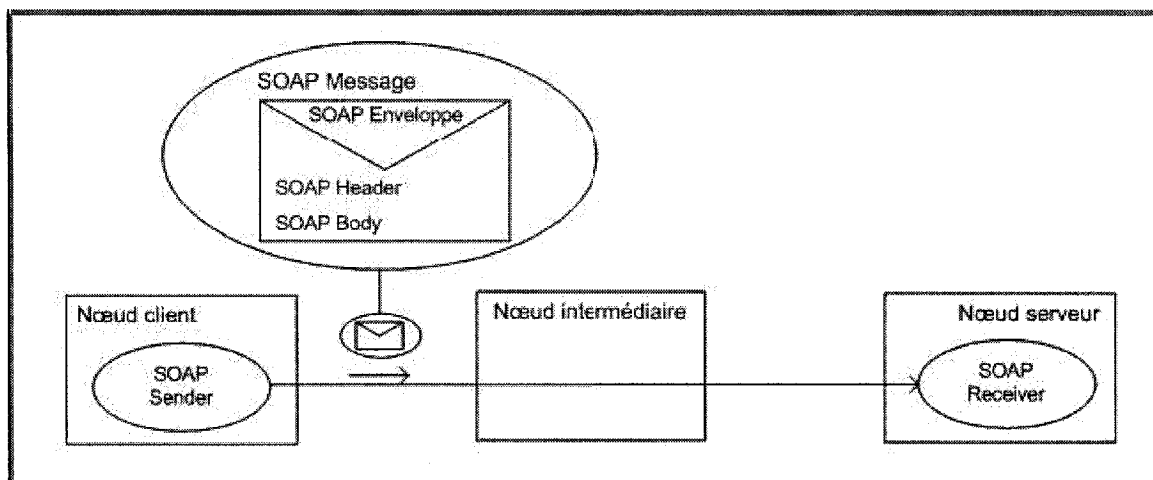


Figure 7.2 : Structure d'un message SOAP

7.1.2.2 Enveloppe SOAP

Un message SOAP est un document XML composé d'un élément obligatoire : l'enveloppe SOAP. Celle-ci définit l'espace de noms précisant la version supportée de SOAP.

L'enveloppe SOAP est constituée d'un en-tête facultatif (*SOAP Header*) et d'un corps obligatoire (*SOAP Body*).

L'en-tête peut contenir plusieurs éléments qui servent à désigner une fonctionnalité du message, comme l'authentification. L'en-tête peut également utiliser des attributs *mustUnderstand*²⁶ et/ou *SOAP actor*²⁷ pour indiquer comment traiter l'entrée et par qui.

Le corps SOAP contient l'information destinée au receveur. Il doit fournir le nom de la méthode invoquée par une requête, les valeurs des paramètres, la valeur de retour. Il doit être identifié par son nom qualifié (URI de l'espace de noms + nom local), et doit

²⁶ L'attribut *MustUnderstand* est utilisé pour indiquer si une entrée d'en-tête est obligatoire ou facultative pour être traitée par le destinataire. Si cette entrée est obligatoire, la valeur de *MustUnderstand* est à 1. Autrement, elle est à 0.

²⁷ Le destinataire d'une entrée d'en-tête (*SOAP Header*) est défini par l'attribut *SOAP actor*. La valeur de cet attribut est un URI. Par exemple, l'URI "<http://schemas.xmlsoap.org/soap/actor/next>" indique que l'élément d'en-tête est destiné à la première entité SOAP qui traite le message. Si cet attribut est absent, alors le SOAP Receiver est le destinataire du message SOAP.

indiquer le jeu d'encodage des entrées. Le corps SOAP peut également contenir des éléments <faults> (retours d'erreurs).

7.1.2.3 SOAP sur HTTP

En plus de l'enveloppe, un message SOAP comporte un en-tête utilisant des notifications HTTP. La structure de ce dernier dépend du type de message : requête ou réponse. Des exemples de requête ou réponse sont montrés aux figures 7.7 et 7.8.

Les requêtes SOAP sont des requêtes HTTP POST. Les applications HTTP doivent utiliser le type "text/xml" lorsque les messages HTTP comprennent des corps SOAP. Lorsque le message est une réponse, l'en-tête SOAP HTTP montre le statut de la réponse. Un code HTTP 2xx (200 par exemple) indique que le récepteur a correctement reçu, compris et accepté le message. Un code HTTP 500 signale une erreur et que le récepteur n'accepte pas le message.

Pour plus d'informations sur le protocole SOAP, se référer aux documents [15] [16].

7.1.3 WSDL

WSDL (Web Services Description Language) est un langage du W3C qui fournit une description XML sur la façon de communiquer en utilisant un service réseau (ou service Web). Plus précisément, WSDL décrit de manière formelle une interface publique d'accès à un service Web et indique la façon de communiquer en utilisant le service Web.

Un service WSDL est composé d'un ensemble d'opérations élémentaires, chacune décrite par un flux de messages échangés entre le client et le service. De plus, WSDL spécifie les protocoles de transport et d'échange des messages.

Un document WSDL est un fichier XML qui dispose d'un élément racine <definitions> spécifiant l'espace de noms de WSDL (<http://schemas.xmlsoap.org/wsdl>) et qui définit

un service Web comme un ensemble de points de terminaison réseau (*endpoints*) ou ports.

Tout document WSDL présente la structure de premier niveau suivante :

```
<wsdl:definitions>
  <wsdl:types>...</wsdl:types>
  <wsdl:message>...</wsdl:message>
  <wsdl:portType>...</wsdl:portType>
  <wsdl:binding>...</wsdl:binding>
  <wsdl:service>...</wsdl:service>
</wsdl:definitions>
```

Figure 7.3: Structure d'un document WSDL

Cette structure présente les éléments principaux d'un document WSDL. Dans la spécification WSDL 1.1 [17], une description WSDL définit d'abord les types utilisés (<wsdl:types>) pour former des messages (<wsdl:message>) associés à des ports (<wsdl:portType>) reliés à des protocoles par des "bindings" (<wsdl:binding>) formant un service Web (<wsdl:service>).

Les éléments <wsdl:types>, <wsdl:message> et <wsdl:portType> correspondent à la partie abstraite du document WSDL. Les éléments <wsdl:binding> et <wsdl:service> représentent la partie concrète (cf. figure 7.4).

L'élément <wsdl:types> contient les définitions de types de données qui peuvent être effectuées à l'aide d'un schéma XML (.xsd). Généralement, cet élément comprend un élément `schema` qui définit les types de données.

WSDL décrit un service Web en termes des messages qu'il envoie et reçoit. Un élément <wsdl:message> contient la définition des données du message : il décrit les noms et types d'un ensemble de champs à transmettre, comme les paramètres d'une invocation, la valeur de retour, ...

Chaque méthode Web est associée à deux messages :

- Un message d'entrée qui décrit les paramètres d'entrée
- Un message de sortie qui décrit les données de retour de la méthode.

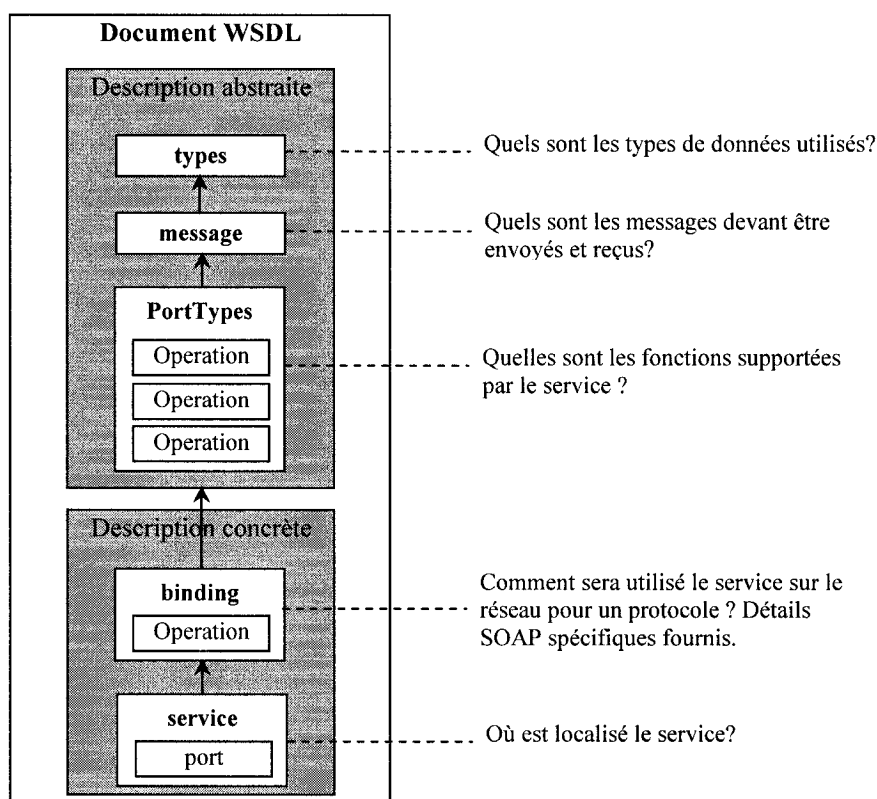


Figure 7.4 : Éléments abstraits et concrets d'un document WSDL

L'élément `<wsdl:portType>` définit un jeu abstrait d'opérations prises en charge par un ou plusieurs points de terminaison. Chaque opération a zéro ou un message en entrée, zéro ou plusieurs messages de sortie ou de fautes. Il existe plusieurs types d'opérations :

- One way : le point de terminaison reçoit un message (`<input>`)
- Request-response : le point de terminaison reçoit un message (`<input>`) et retourne un message corrélé (`<output>`) avec un ou plusieurs messages d'erreurs (`<fault>`)
- Solicit-response : le point de terminaison envoie un message (`<output>`) et reçoit un message corrélé (`<input>`) avec un ou plusieurs messages d'erreurs (`<fault>`)
- Notification : le point de terminaison envoie un message de notification (`<output>`).

Les champs des messages constituent les paramètres (in, out, inout) des opérations.

L'élément `<wsdl:binding>` spécifie le protocole de transport et le format des messages SOAP pour un type de port (*portType*) particulier. Un *portType* peut avoir plusieurs "bindings". Un "binding" permet de définir la liaison WSDL/SOAP.

Finalement, un `<wsdl:service>` regroupe un ensemble de points de terminaison (ou port) associés. Un port associe une adresse réseau à un "binding".

7.1.4 Service Web et .NET

Le .NET Framework fournit la prise en charge des services Web permettant ainsi de développer, de découvrir, de déboguer, de déployer et de consommer des services Web. Visual Studio facilite également pour les développeurs la publication et la localisation des services Web dans UDDI. Dans la page de démarrage de Visual Studio, les développeurs peuvent publier des informations sur leurs services Web directement dans UDDI. À mesure que les développeurs créent de nouveaux services, ils peuvent les porter à la connaissance de tous les utilisateurs de l'entreprise, directement à partir de Visual Studio. En se connectant à UDDI via la boîte de dialogue « Ajouter une référence Web », un développeur peut localiser les services Web disponibles.

Les services Web dans .NET utilisent des normes ouvertes telles que le HTTP, le SOAP, le XML et WSDL.

7.2 Création d'un service Web

Nous allons tenter de créer, avec .NET, un service Web mettant à la disposition d'applications clientes les méthodes développées dans le chapitre 6.

L'activation d'un service Web avec ASP.NET implique :

- la création d'un fichier portant l'extension `.asmx`
- la déclaration d'un service Web dans ce fichier – ou dans un fichier séparé
- la définition des méthodes du service Web.

7.2.1 Déclaration de service Web

Pour créer un service Web, il faut placer la directive `@WebService` au début du fichier portant l'extension `.asmx`. La présence du fichier `asmx` et de la directive `@WebService` met en corrélation l'adresse URL du service Web et son implémentation.

Ensuite, il suffit d'implémenter le langage de programmation et la classe de service Web définissant les méthodes et les types de données auxquels peuvent avoir accès les clients des services Web.

Par exemple, si le langage utilisé est C# et que le nom de la classe de service Web est *Service*, alors la directive peut s'écrire comme suit :

```
<%@WebService Language="C#" Class="Service"%>
```

La classe de service Web définie peut être directement incluse dans le fichier `.asmx` ou alors être dans un fichier séparé. Dans ce dernier cas, le fichier séparé doit être compilé dans un assemblage. Et la directive `@WebService` est la seule ligne du fichier `.asmx`; elle spécifie alors le nom de l'assemblage contenant la classe de service Web.

L'attribut `WebService` peut optionnellement être appliqué à la classe de service Web, comme suit :

```
[WebService(Namespace = "http://tempuri.org/")]
public class Service
{
    // méthodes
}
```

Cela permet de définir l'espace de noms XML par défaut pour le service Web ainsi qu'une chaîne décrivant le service. Si cet espace de noms n'est pas modifié, l'espace de noms par défaut est alors <http://tempuri.org/>.

Remarque : dans .NET, le fichier implémentant la classe de service Web doit faire référence à l'espace de noms pour les services Web donné par *System.Web.Services*.

7.2.2 Définition des méthodes du service Web

Pour déclarer des méthodes de service Web à l'aide d'ASP .NET, deux conditions sont requises :

- les méthodes de la classe implémentant le service Web doivent être publiques
- l'attribut `WebMethod` (`[WebMethod]` en C#) doit être appliqué à toutes les méthodes publiques avec lesquelles il est souhaité de communiquer sur le Web. On parle alors de méthodes de service Web.

```
public Service ()
{
    [WebMethod]
    public int nombre(String fileName, String element)
    {
        //cf. fig 6.1 pour le contenu de la methode
    }
}
```

Ces méthodes constituent une partie importante de l'infrastructure de messagerie utilisée par les services Web. Nous savons qu'un client et un serveur Web communiquent à travers un système de messages SOAP. Le client envoie une requête SOAP à un serveur Web et par la suite, une méthode du service Web retourne une réponse SOAP.

Les services Web définissent le type de messages qu'ils acceptent à l'aide d'opérations, comme cela est défini dans le langage WSDL. Ces opérations correspondent à chacune des méthodes d'un service Web.

Les messages transférés sont des documents XML ; par conséquent, les données – pour les méthodes de service Web – transmises sur le réseau doivent être sérialisées en XML.

Le code contenant l'ensemble des méthodes Web définies est disponible dans le fichier **Service.cs** du CD-ROM.

7.2.3 Génération automatique de document WSDL

En suivant toutes ces étapes, nous créons un service Web nommé **Service.asmx**. Ce service Web est placé sur un serveur appelé par exemple **localhost**²⁸, dans un répertoire

²⁸ Le service Web est disponible en local, sur l'ordinateur.

virtuel appelé **cim2emtp**. Les méthodes Web associées à ce service sont celles définies pour l'interface activeX, dans la section 6.1.2. Après compilation du code créé, l'infrastructure des services Web dans .NET utilise ce code pour générer de façon dynamique un fichier WSDL. Ce document WSDL est consultable d'ailleurs à l'adresse <http://localhost:1151/cim2emtp/Service.asmx?WSDL>.

En entrant dans un navigateur l'URL <http://localhost:1151/cim2emtp/Service.asmx>, une page s'affiche, indiquant les méthodes prises en charge par le service Web (celles marquées par l'attribut **WebMethod** – cf. figure 7.5)

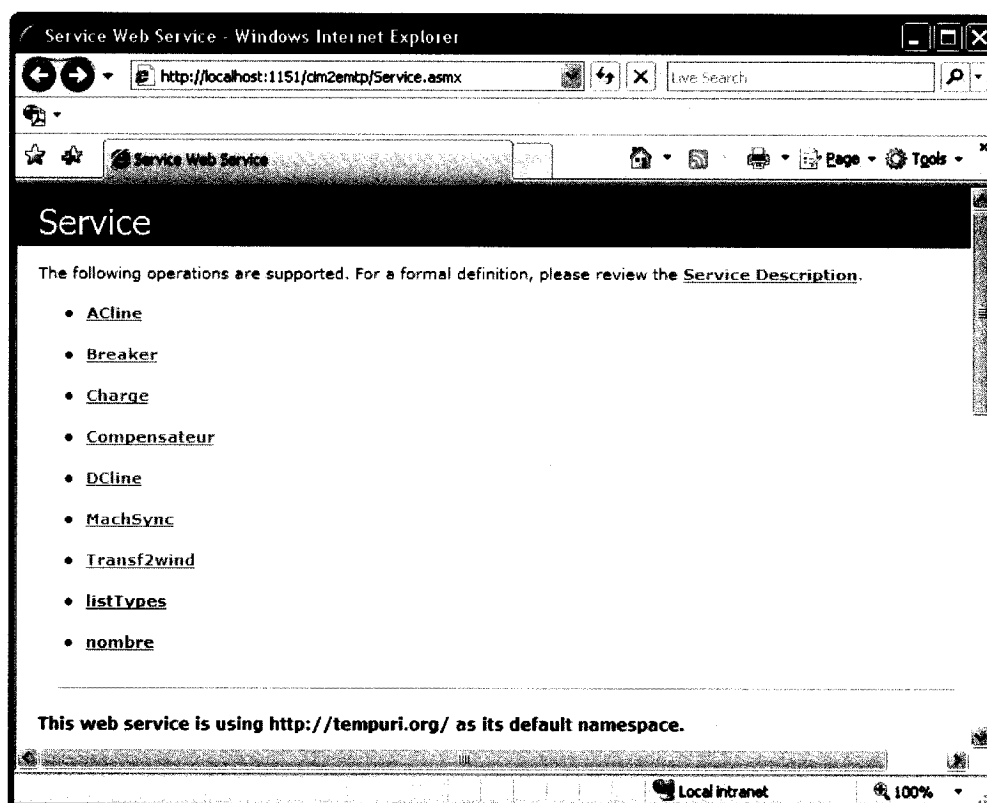


Figure 7.5 : Liste des méthodes du service Web créé

En cliquant sur l'une des méthodes ou opérations, il est possible de voir un exemple de requête et de réponse SOAP du service Web. Également l'information sur les paramètres d'entrée et de retour de la méthode est fournie. Cette information sera nécessaire à la

conception d'une interface dans EMTP-RV (avec JavaScript) pour consommer ce service Web.

7.2.4 Test du service Web créé

Pour voir si le service Web créé fonctionne bien, nous pouvons le tester directement sur Internet en lui assignant comme fichier d'entrée le document CIM-XML du réseau à 6 barres (res_PM.xml).

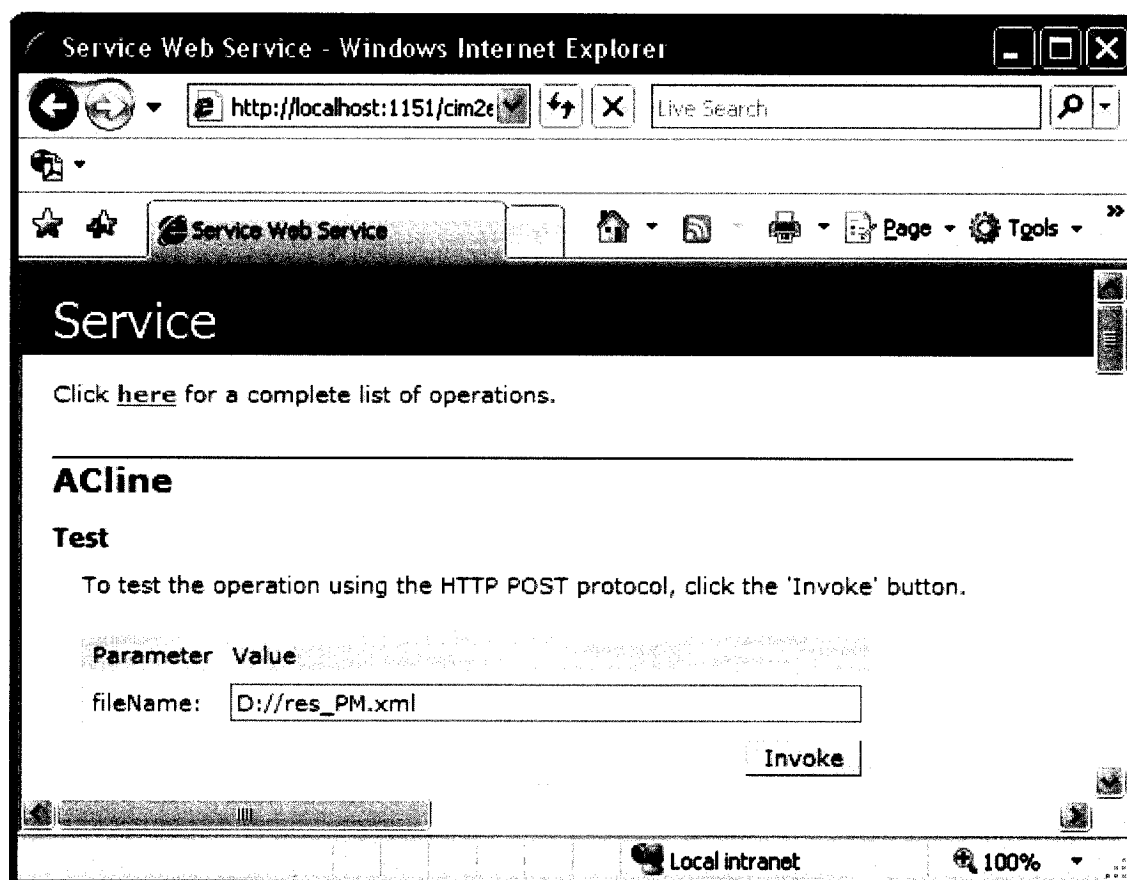


Figure 7.6 : Test de la méthode Web ACline

La méthode Web ACline retourne alors le flux XML suivant :

```
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://tempuri.org/">TLM[0].name= 'LINE1';TLM[0].nbphases =
'3';TLM[0].ctransposed = 'ON';TLM[0].distortioncheck = 'OFF';TLM[0].lccheck =
'ON';TLM[0].zvcheck = 'OFF';TLM[0].ztcheck = 'OFF';TLM[0].lc_runits=
';TLM[0].lc_cunits= 'uF';TLM[0].lc_lunits= 'mH';TLM[0].linelen=
'100';TLM[0].propag_grid=
'100\t0.316764\t3.22209182289542\t0.00786994667765574\n100\t0.0243404\t0.923800
012715551\t0.0126000315863469';TLM[0].pin1= 'CN2';TLM[0].pin2= 'CN3';
```

```

TLM[1].name= 'LINE2';TLM[1].nbphases = '3';TLM[1].ctransposed =
'ON';TLM[1].distortioncheck = 'OFF';TLM[1].lccheck = 'ON';TLM[1].zvcheck
='OFF';TLM[1].ztcheck = 'OFF';TLM[1].lc_runits= '';TLM[1].lc_cunits=
'uF';TLM[1].lc_lunits= 'mH';TLM[1].linelen= '75';TLM[1].propag_grid=
'75\t0.316764\t3.22209182289542\t0.00786994667765574\n75\t0.0243404\t0.92380001
2715551\t0.0126000315863469';TLM[1].pin1= 'CN3';TLM[1].pin2= 'CN4';
TLM[2].name= 'LINE3';TLM[2].nbphases = '3';TLM[2].ctransposed =
'ON';TLM[2].distortioncheck = 'OFF';TLM[2].lccheck = 'ON';TLM[2].zvcheck
='OFF';TLM[2].ztcheck = 'OFF';TLM[2].lc_runits= '';TLM[2].lc_cunits=
'uF';TLM[2].lc_lunits= 'mH';TLM[2].linelen= '120';TLM[2].propag_grid=
'120\t0.316764\t3.22209182289542\t0.00786994667765574\n120\t0.0243404\t0.923800
012715551\t0.0126000315863469';TLM[2].pin1= 'CN3';TLM[2].pin2= 'CN6';</string>

```

L'espace de noms de ce flux XML est bien celui défini par défaut dans le serveur Web (<http://tempuri.org/>). Le texte contenu dans l'élément XML correspond bien à un tableau de trois objets TLM, encapsulé dans un *String*. Le résultat retourné par cette méthode Web est similaire à celui retourné par la méthode ACline exposée par l'activeX (cf. section 6.2.1).

Pour les autres méthodes Web, les résultats sont concordants avec ceux obtenus avec les méthodes de l'objet activeX.

Donc le serveur Web fonctionne correctement.

7.3 Consommation du service Web

L'application cliente devant consommer le service Web créé précédemment est une application JavaScript de EMTP-RV.

Afin que ce client et le service Web puissent communiquer à l'aide de messages SOAP, il faut créer un proxy. Ce proxy sert de passerelle entre l'application JavaScript dans EMTP-RV et le serveur Web créé. Il est responsable de la formation des messages SOAP et de leur envoi sur le réseau.

Pour créer la classe proxy, nous allons utiliser la technologie AJAX qui est décrite dans la section suivante. Celle-ci a permis de créer une librairie de classes en vue de développer une classe de proxy.

7.3.1 Technologie AJAX

AJAX signifie Asynchronous JavaScript And XML. AJAX évoque l'utilisation conjointe d'un ensemble de technologies, entre autres :

- DOM (cf. section 2.1.1) et JavaScript pour afficher et interagir dynamiquement avec l'information présentée
- L'objet XMLHttpRequest pour échanger les données avec le serveur Web
- XML

Pour pouvoir configurer les messages SOAP à envoyer au serveur Web, nous allons nous intéresser et utiliser l'objet **XMLHttpRequest**²⁹. Il s'agit d'un objet COM local qui dépend du navigateur (Internet Explorer, Mozilla, ...). Les objets XMLHttpRequest permettent de récupérer toutes sortes de données, notamment des données XML.

Pour effectuer une requête, il faut tout d'abord déclarer une instance de cet objet. Sous Internet Explorer, cela peut se faire par exemple comme suit :

```
var xmlHttp = new activeXObject("Microsoft.XMLHTTP")
```

Ensuite, il faut initialiser la requête avec la méthode **open** à laquelle on spécifie la méthode de transmission des données, l'URL et le mode de transmission de la requête³⁰ (synchrone ou asynchrone).

Par la suite, la requête peut être exécutée avec la méthode **send**.

L'exemple ci-dessous montre une requête sur le fichier WSDL du service Web créé plus tôt. La transmission se fait de façon synchrone.

```
var url = "http://localhost/cim2emtp /Service.asmx?wsdl"
xmlHttp.open("GET", url, false);
xmlHttp.send(null);
```

²⁹ XMLHttpRequest a tout d'abord été développé par Microsoft, en tant qu'objet ActiveX pour Internet Explorer 5. Il est disponible depuis 1998. Il a ensuite été repris et implémenté successivement sous Mozilla 1.0 (mai 2002), Safari 1.2 (février 2004), Konqueror 3.4 (mars 2005) et dernièrement Opera 8.0 (avril 2005). En raison de ces implémentations plutôt récentes, l'objet n'est pas supporté par les navigateurs dits de "vieille génération".

³⁰ Le mode de transmission de la requête peut être synchrone ou asynchrone. S'il est synchrone, le paramètre à rentrer dans la méthode **open** est *false*. Autrement, *true*.

Lorsque la requête se termine, des informations sont reçues du serveur. Ces informations sont contenues dans la variable `xmlHttp.responseText` ou `xmlHttp.responseXML`, selon que le résultat est au format texte ou XML.

7.3.1.1 Synchrone ou asynchrone

Lorsque le mode synchrone est choisi, alors tant que le résultat de la requête n'est pas disponible, le script est en pause et le navigateur bloqué. Avec un bon débit et/ou peu de données à transmettre, il est possible de s'accommoder avec ce mode. Toutefois, il peut s'avérer gênant pour l'utilisateur qui voit son navigateur figé. Aussi, il existe le mode asynchrone dans lequel, après avoir envoyé la requête avec la méthode *send*, le script poursuit son exécution sans bloquer le navigateur. Dans ce mode, la propriété ***onreadystatechange*** peut être utilisée pour avertir lorsque la requête a abouti. Il s'agit en fait d'une fonction qui fait appel à une autre propriété de `xmlHttp` qu'est ***readyState***. Cette propriété prend successivement cinq valeurs récapitulées dans l'annexe I. A chaque changement d'état de *readyState*, la fonction *onreadystatechange* est appelée. Mais c'est l'état 4 qui est intéressant comme cet état est atteint lorsque les données sont complètement accessibles. L'exemple ci-dessous montre une utilisation de cette propriété.

```
var url = "http://localhost/cim2EMTP-RV/Service.asmx?wsdl"
xmlHttp.open("GET", url, async);
if(async)
{
    xmlHttp.onreadystatechange = function()
    {
        if(xmlHttp.readyState == 4)
        {
            alert("Opération terminée!")
        }
    }
}
xmlHttp.send(null);
```

Une autre propriété est ***status*** et elle représente le code HTTP retourné par la requête. Les valeurs de code que peut retourner ***status*** sont présentées en annexe I. Deux valeurs

sont intéressantes à noter : le code 200 qui montre que la requête a bien abouti et le code 500 équivalent à une erreur.

7.3.1.2 Modes de transmission des données

XMLHttpRequest permet de réaliser une requête en transmettant des données. Les données sont transmises via l'une des deux méthodes : GET ou POST. La méthode choisie doit être spécifiée lors de l'initialisation de la requête avec la méthode *open*.

Si la méthode sélectionnée est GET, les données à transmettre sont concaténées à l'URL et si la méthode est POST, elles sont envoyées via la fonction *send*. Dans ce dernier cas, il faut préciser l'encodage en spécifiant l'en-tête adéquat avec la propriété *setRequestHeader*.

7.3.2 Développement de la classe proxy

Du côté client, la classe proxy JavaScript est générée à partir du WSDL.

Le schéma de la figure 7.7 montre les interactions entre l'application cliente JavaScript et le serveur Web ainsi que les différentes étapes de développement de la classe proxy appelée **SOAPClient**. Cette classe proxy est une fonction JavaScript.

La méthode principale de la classe SOAPClient est la fonction *invoke* qui prend comme paramètres d'entrée :

- l'adresse URL du service Web
- le nom de la méthode Web
- les valeurs des paramètres d'entrée de la méthode (*param*)
- le mode d'appel (synchrone ou asynchrone)
- la méthode de retour *callback* invoquée (en mode asynchrone) dès réception de la réponse du serveur (optionnelle pour les appels synchrones).

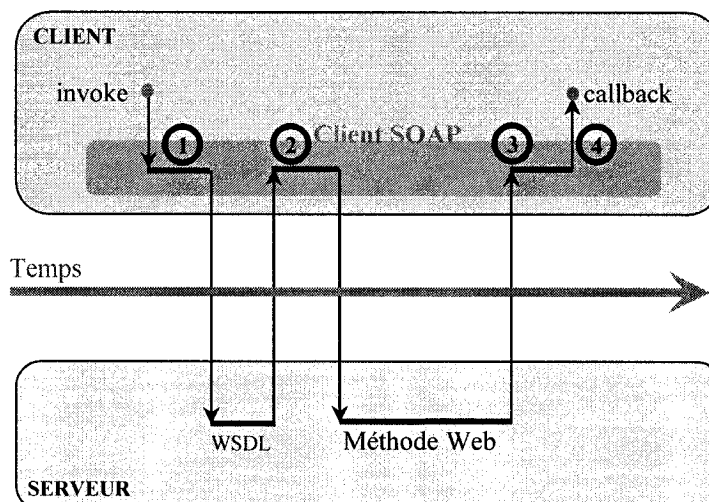


Figure 7.7: Processus d'appel asynchrone de méthodes Web

La méthode ***invoke*** exécute les opérations suivantes (cf. fig. 7.7) :

1. elle va chercher le document WSDL
2. elle prépare et envoie une requête **SOAP 1.1** au serveur
3. elle traite la réponse du serveur
4. si l'appel est asynchrone, la méthode ***callback*** est invoquée, sinon, elle retourne l'objet fourni par la requête.

Nous allons passer en revue le processus et analyser le code.

Afin d'utiliser la méthode ***invoke***, il faut pouvoir spécifier la valeur du paramètre d'entrée de la méthode considérée. C'est l'objet de la classe `SOAPClientParameters`.

Cette classe a deux méthodes :

- la fonction ***add*** qui attache à chaque nom de paramètre sa valeur.
- La méthode ***toXml*** qui fait la sérialisation XML des requêtes SOAP.

```
function SOAPClientParameters()
{
    var _pl = new Array();
    this.add = function(name,value)
    {
        _pl[name] = value;
        return this;
    }
}
```

```

    }

    this.toXml = function()
    {
        var xml = "";
        for (var p in _pl)
        {
            if (typeof(_pl[p]) != "function")
            {
                xml += "<" + p + ">" + _pl[p].toString().replace(/&/g, "&amp;")
                .replace(/</g, "&lt;").replace(/>/g, "&gt;") + "</" + p + ">";
            }
        }
        return xml;
    }
}

```

Le code ci-dessous décrit le contenu de la méthode principale *invoke*.

```

SOAPClient.invoke = function(url,method,param,async,callback)
{
    return SOAPClient._loadWsdL(url,method,param,async,callback);
}

```

La procédure d'invocation commence en faisant appel à la méthode *_loadWsdL* qui charge le fichier WSDL. Le code de cette méthode est détaillé ci-dessous :

```

1 SOAPClient._loadWsdL = function(url,method,param,async,callback)
2 {
3     var xmlHttp = SOAPClient._getXmlHttp();
4     xmlHttp.open("GET", url + "?wsdl", async);
5     if(async)
6     {
7         xmlHttp.onreadystatechange = function()
8         {
9             if(xmlHttp.readyState ==4)
10            {
11                SOAPClient._onLoadWsdL(url,method,param,async,callback,xmlHttp);
12            }
13        }
14    }
15    xmlHttp.send(null);
16    if(!async)
17    {
18        return SOAPClient._onLoadWsdL(url,method,param,async,callback,xmlHttp);
19    }
20 }

```

La méthode va chercher le fichier WSDL en utilisant la méthode GET de l'objet XMLHttpRequest (ligne 4). La méthode *_getXmlHttp* (ligne 3) crée une instance de l'objet XMLHttpRequest suivant la version et le type de navigateur.

Dès que la réponse du serveur sous format XML (responseXML) est disponible, la méthode *_onLoadWsdL* est invoquée.

```
SOAPClient._onLoadWsdL= function(url,method,param,async,callback,req)
{
    var wsdl = req.responseXML;
    return SOAPClient._sendSoapRequest(url,method,param,async,callback,wsdl);
}
```

Cette méthode récupère le document WSDL puis fait exécuter la méthode *_sendSoapRequest*.

```
1 SOAPClient._sendSoapRequest = function(url,method, param,async,callback,wsdl)
2 {
3     var ns =
4     (wsdl.documentElement.attributes["targetNamespace"]+"=="+"undefined") ?
5     wsdl.documentElement.attributes.getNamedItem("targetNamespace").nodeValue :
6     wsdl.documentElement.attributes["targetNamespace"].value;
7     // build SOAP request
8     var sr = "<?xml version=\"1.0\" encoding=\"utf-8\"?>" + "<soap:Envelope " +
9             "xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" " +
10            "xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\" " +
11            "xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">" +
12            "<soap:Body>" + "<" + method + " xmlns=\"" + ns + "\">" +
13            parameters.toXml() + "</"+method + "></soap:Body></soap:Envelope>";
14     // send request
15     var xmlHttp = SOAPClient._getXmlHttp();
16     xmlHttp.open("POST", url, async);
17     var soapaction = "\""+(ns.lastIndexOf("/") != ns.length-1) ? ns+"/" : ns) +
18             method + "\"";
19     xmlHttp.setRequestHeader("SOAPAction", soapaction);
20     xmlHttp.setRequestHeader("Content-Type", "text/xml; charset=utf-8");
21     if (async)
22     {
23         xmlHttp.onreadystatechange = function()
24         {
25             if (xmlHttp.readyState == 4)
26                 SOAPClient._onSendSoapRequest(method, async, callback, wsdl,
27                 xmlHttp);
28         }
29     }
30     xmlHttp.send(sr);
31     if (!async)
32         return SOAPClient._onSendSoapRequest(method, async, callback, wsdl,
33         xmlHttp);
34 }
```

L'espace de noms du service Web (par défaut <http://tempuri.org>) est extrait du document WSDL (lignes 3 à 6). Ensuite, une requête SOAP 1.1 est créée (lignes 8 à 20) puis envoyée (ligne 30).

Le format de la requête SOAP 1.1 pour la méthode DCline par exemple est montré à la figure 7.8.

```
POST /webServer/cim2emtp/Service.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/DCline"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DCline xmlns="http://tempuri.org/">
      <fileName>string</fileName>
    </DCline>
  </soap:Body>
</soap:Envelope>
```

Figure 7.8 : Exemple de format de requête SOAP 1.1

Une fois obtenue, la réponse sous format XML (cf. Fig. 7.9 pour la méthode DCline par exemple) est traitée par la méthode *_onSendSoapRequest*.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DClineResponse xmlns="http://tempuri.org/">
      <DClineResult>string</DClineResult>
    </DClineResponse>
  </soap:Body>
</soap:Envelope>
```

Figure 7.9 : Exemple de format de réponse du serveur avec le protocole SOAP 1.1

Cette méthode utilise les DOM (*Document Object Model*, cf. section 2.1.1) pour parcourir le document XML et extraire l'information retournée par le serveur Web sous forme de chaîne de caractères (*String*).

Le code complet de la classe proxy est disponible dans le fichier "*test_Web.dwj*" rajouté au CD-ROM.

7.3.3 Test de la classe proxy

Nous allons tester la classe proxy créée en utilisant le réseau test à 6 barres.

Nous spécifions :

- le fichier XML test (res_PM.xml) comme paramètre
- l'adresse URL du serveur Web
- la méthode Web à appeler (par exemple la méthode *Charge*)
- le mode de transmission des données (mode synchrone $\Rightarrow$ variable *async* = false)

```
var nom_fichier="D:\\res_PM.xml";
var pl = new SOAPClientParameters();
pl.add("fileName",nom_fichier);

url="http://localhost:1151/cim2emtp/Service.asmx";
method="Charge";

async=false;

if (async)
{
    function callback (r)
    {
        resultat=r;
        return resultat;
    }
    SOAPClient.invoke(url,method,pl,async,callback);
}

if(!async)
{
    resultat= SOAPClient.invoke(url,method,pl,async,callback);
}
```

Le résultat retourné est le même que celui retourné par la méthode *Charge* de l'interface activeX (cf. section 6.2.1). Il s'agit d'un tableau de deux objets PQload, sous format *String*.

Donc la classe proxy fonctionne bien.

7.4 Résultats

A partir de là, la procédure pour charger les données CIM-XML retournées par les diverses méthodes Web dans EMTPWorks est la même que celle définie pour l'interface ActiveX. En fait, ce qui change, c'est juste la démarche utilisée pour importer ces données dans l'application JavaScript. Dans la solution précédente, nous avons utilisé un **activexObject** alors que dans la présente solution, nous passons par la classe proxy **SOAPClient**. Dans les deux cas, le résultat retourné par les méthodes sont des tableaux d'objets sous format *String*.

La même procédure que celle décrite à la section 6.2.2 pour le traitement des données est suivie. Le fichier *test_Web.dwj* rajouté dans le CD-ROM contient le code intégral de la procédure suivie.

Nous parvenons aux mêmes résultats obtenus avec l'interface activeX :

- Avec comme document CIM-XML celui du réseau à 6 barres, nous importons dans EMTPWorks le circuit de la figure 6.3, avec des données dans les objets EMTP-RV conformes aux données contenues dans le fichier CIM-XML
- En considérant le réseau à 60 barres, nous parvenons également à importer les mêmes éléments du circuit en annexe G.

7.5 Conclusion

Dans ce chapitre, nous avons exploré la technologie des services Web. Un serveur Web a d'abord été créé en utilisant .NET. Ce serveur Web rend accessibles depuis une adresse Internet des méthodes qui permettent d'extraire les données contenues dans les documents CIM-XML. Ces méthodes sont identiques à celles utilisées dans la solution précédente. Pour déployer ces méthodes dans une application JavaScript de EMTP-RV, nous avons créé une passerelle (classe proxy) entre JavaScript et le serveur Web en utilisant la technologie appelée AJAX.

À la suite de cela, nous sommes parvenus non seulement à extraire les données contenues dans des documents CIM-XML mais également à les importer de façon visuelle dans EMTPWorks.

Un aspect de comparaison de cette solution par rapport à la solution basée sur l'interface activeX, c'est qu'il n'est pas nécessaire qu'une application cliente dispose de .NET pour faire appel aux méthodes exposées par le serveur Web. L'environnement .NET a été utilisé juste pour créer le serveur Web. Pour une application cliente, seule une connexion Internet suffit.

Dans le cadre de cette étude, nous avons placé le serveur Web créé en local sur un ordinateur (domaine = *localhost*). Mais il est bien possible de créer le serveur Web et de le placer à une adresse Web publique, accessible de n'importe quel ordinateur connecté à Internet.

L'avantage de l'utilisation des services Web, c'est que ces derniers sont basés sur des normes et protocoles ouverts. Généralement dérivées du XML, ces normes permettent d'uniformiser la présentation des services Web, rendant ainsi leur accès transparent à tout type de plate-forme. De plus, les services Web peuvent fonctionner au travers des pare-feux installés sur les ordinateurs.

Toutefois, la principale inquiétude soulevée par l'utilisation des services Web concerne les aspects de sécurité. Comme il s'agit d'échanger des informations à travers le Web, la gestion de la confidentialité, les vérifications de droits d'accès sont autant d'aspects qui n'ont pas été prévus lors de la création de services Web.

CONCLUSION

Le travail effectué dans le cadre de ce projet de recherche a comporté deux volets : tout d'abord décrire le modèle d'information commun (CIM) et puis ensuite étudier la faisabilité que ce modèle puisse être utilisé par EMTP-RV pour importer des données.

Le modèle CIM fournit un vocabulaire commun pour la description des ressources d'un système électrique, de leurs attributs et définit les relations entre ces ressources. Il est défini comme un ensemble de classes regroupées dans des paquets.

Il existe diverses normes autour du modèle CIM : la norme à laquelle fait référence le modèle CIM utilisé dans le projet est la partie 301 de la série CEI 61970 établie pour la communication avec les centres de commande. Les opérations de distribution et la communication en temps réel des données d'une sous-station font l'objet respectivement des normes CEI 61968 et CEI 61850. Actuellement, des efforts sont faits pour essayer d'harmoniser les modèles contenus dans ces trois normes.

La structure du modèle CIM de base (CEI 61970-301) est disponible sous la forme de diagramme UML de Rational Rose. Le format d'échange du modèle CIM se présente sous la forme d'un document XML. Ce dernier, encore appelé document CIM-XML, comporte des éléments CIM et des éléments empruntés à la méthodologie RDF.

Les données de réseau que nous avons transférées vers EMTP-RV sont initialement sous format CIM-XML. La technique de sérialisation XML a été utilisée pour extraire ces données du document CIM-XML.

Pour le transfert de ces données vers le format de EMTP-RV, il a fallu trouver des correspondances entre les objets CIM et les objets EMTP-RV. Ce travail a nécessité de faire plusieurs hypothèses sur certains paramètres non représentés dans le modèle CIM,

comme par exemple le nombre de pôles d'une machine synchrone. De plus, à la différence de EMTP-RV, le modèle CIM ne permet la représentation des éléments d'un système électrique phase par phase.

Trois solutions ont été développées en utilisant la plate-forme .NET de Microsoft.

Une première solution a consisté à générer le format netlist de tous les objets CIM d'un document CIM-XML. Le fichier netlist résultant sera en fait utilisé pour lancer la simulation du réseau importé. Nous sommes parvenus à générer le fichier netlist pour un réseau test à 6 barres et un autre réseau à 60 barres. L'inconvénient avec cette solution, c'est que non seulement, elle ne permet pas la visualisation graphique du circuit importé mais également, il va être très délicat d'apporter des modifications futures au netlist généré, comme ce dernier suit une syntaxe et une structure bien spécifiques.

La deuxième solution a utilisé un serveur COM (activeX) créé avec .NET qui met à la disposition d'une application Javascript de EMTP-RV des méthodes retournant des objets, « clones » des objets EMTP-RV correspondant aux objets CIM. Ces objets retournés par les méthodes contiennent les données CIM-XML à importer et l'intérêt de les prendre comme des « clones » des objets EMTP-RV, c'est qu'il suffit de « mapper » ces objets avec les objets EMTP-RV pour pouvoir les représenter graphiquement dans EMTPWorks. Cette solution a permis d'avoir une visualisation graphique du réseau importé à 6 barres. Également, certains éléments du réseau à 60 barres ont pu être importés de façon correcte. Une limitation pour cette solution, c'est qu'elle utilise un serveur COM créé avec .NET. Pour qu'une application JavaScript puisse faire appel aux méthodes exposées par ce serveur COM, il faut placer ce serveur COM dans le registre de base de l'ordinateur, ce qui ne peut se faire qu'avec .NET. Cela signifie qu'un autre utilisateur ne peut exploiter cette solution qu'à la condition qu'il ait .NET installé sur son ordinateur.

Une dernière solution, plus sophistiquée, utilise la technologie des services Web. Ces derniers permettent à diverses applications de dialoguer à distance via Internet et sont basés sur des normes ouvertes telles que XML, HTTP et SOAP. Les méthodes exposées par le service Web créé sont similaires à celles de l'interface COM. La technologie AJAX est utilisée pour récupérer les données CIM et créer les émules des objets EMTP-RV comme dans la solution précédente. Avec cette solution, nous sommes parvenus aux mêmes résultats qu'avec la deuxième solution, avec l'importation réussie du réseau test à barres et l'importation partielle du réseau à 60 barres. Un avantage de cette solution par rapport à la précédente, c'est qu'elle n'est pas tributaire de .NET. Par contre, comme les échanges se font via Internet, les questions de sécurité peuvent se poser.

Finalement, les deux dernières solutions permettent de transférer vers EMTP-RV des modèles de réseaux sous format CIM-XML et de les représenter graphiquement.

En plus d'avoir démontré la compatibilité (dans un sens) du logiciel EMTP-RV avec le modèle CIM, le travail fait dans le cadre de ce projet présente d'autres intérêts :

- la démarche suivie et le travail fait pour rechercher la compatibilité de EMTP-RV avec le modèle CIM pourraient s'appliquer à tout autre logiciel de simulation des transitoires électromagnétiques
- le rapport présente une description de tous les concepts utilisés, toutes les procédures et méthodes suivies pour parvenir aux solutions obtenues. Ce document fournit ainsi un bon support pour qui souhaiterait reprendre ou continuer ce travail
- ce projet a mis en relief les différences entre les modèles de représentation CIM et EMTP-RV. Le modèle de base CIM (partie 301 de la série CEI 61970) a été bâti pour l'interopérabilité des logiciels de stabilité. Un impact futur de cette étude serait d'amener à élaborer une norme dédiée aux applications de simulation des transitoires électromagnétiques.

Le travail déjà fait dans ce projet offre plusieurs pistes de suite :

- pour cette étude de faisabilité, la recherche de correspondances entre les objets CIM et les objets EMTP-RV a été limitée à quelques éléments du paquet « Wires » du modèle de base CIM. On retrouve dans ce paquet des classes représentant tous les éléments généralement présents dans un réseau électrique. Une continuité intéressante du projet serait d'étendre la recherche de correspondances à tous les objets du modèle CIM
- pour compléter l'étude de la compatibilité de EMTP-RV avec le modèle CIM, une étude de faisabilité pourrait être menée pour la conversion des données de circuits EMTP-RV sous format CIM-XML.

RÉFÉRENCES

DOCUMENTS

- [1]. “Common Information Model (CIM): CIM 10 Version”, Final Report – D. Becker, November 2001
- [2]. “Unified Modeling Language: Superstructure” version 2.0- OMG, August 2005
<http://www.omg.org/docs/formal/05-07-04.pdf>
- [3]. “RDF Vocabulary Description Language 1.0: RDF Schema” – W3C Recommendation, Feb. 2004 – Dan Brickley, R. V. Guha
<http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>
- [4]. “Resource Description Framework (RDF) Model and Syntax Specification” – W3C Recommendation, Feb. 1999 – Ora Lassili, Ralph R. Swick
<http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>
- [5]. “eXtensible Markup Language (XML 1.0 (Third Edition))” – W3C Recommendation, Feb. 2004 – T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau
<http://www.w3.org/TR/REC-xml/>
- [6]. “A Discussion of the Relationship Between RDF-Schema and UML” – W3C, Aug. 1998 – Walter W. Chang
<http://www.w3.org/TR/1998/NOTE-rdf-uml-19980804/>
- [7]. “Knowledge Representation for Power System Modelling” A. deVos, C. T. Rowbotham – Langdale Consultants – PICA 2001
<http://www.langdale.com.au/PICA/KRforPSM.pdf>
- [8]. “Simplified RDF Syntax For Power System Model Exchange” – A. deVos – Langdale Consultants, Nov. 2000
- [9]. “Document Object Model (DOM) Level 2 Core Specification” –version 1.0 – W3C Recommendation, Nov. 2000 – WA. Le Hors, G. Nicol, L. Wood, M. Champion, S. Byrne

<http://www.w3.org/TR/DOM-Level-2-Core/>

- [10]. "XML DOM Tutorial" <http://www.w3schools.com/dom/default.asp?output=print>
- [11]. "XML Schema" W3C, 12 sept.2005
<http://www.w3.org/2001/XMLSchema>
- [12]. "Interoperability Fundamentals"
http://msdn.microsoft.com/library/en-us/dnpag/html/jdni_ch03.asp?frame=true
- [13]. "EDF Feedback on CIM Standard" – E. Lambert, A. Maizener, EPRI CIM/GID International Conference, Nov. 2005
- [14]. "Midwest ISO CIM/XML EMS Model Data Exchange Issues
- [15]. "Les services Web", Jeremy Fierstone – SAR5 – Novembre 2005" – D. Dieser, EPRI CIM/GID International Conference, Nov. 2005
- [16]. "Simple Object Access Protocol (SOAP) 1.1" - W3C Note 08 May 2000 – D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, D. Winer
<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [17]. "Web Services Description Language (WSDL) 1.1" - W3C Note 15 March 2001 – E. Christensen, F. Curbera, G. Meredith, S. Weerawarana
<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- [18]. "Translating CIM-XML Power System Data to a Proprietary Format for System Simulation" – A. W. McMorran, G. A. Ault, I. M. Elders, C. E. T. Foote, G. M. Burt, J. R. McDonald – IEEE, 2004
- [19]. "Semantic interoperability within the Power Systems Domain" – M. Usler, IHIS, Germany, 2005
- [20]. http://www.emtp.com/software/emtp_rv.html
- [21]. "EMTP-EMTPWorks: Quick overview" – J. Mahseredjian, Ecole Polytechnique de Montréal, Nov. 2005

NORMES DE RÉFÉRENCE

Norme CEI 61970-301

Energy management system application program interface (EMS-API) - Part 301:
Common Information Model (CIM) base

Norme CEI 61970-501

Energy management system application program interface (EMS-API) - Part 501:
Common Information Model Resource Description Framework (CIM RDF) schema

Norme CEI 61970-503 (draft)

Energy management system application program interface (EMS-API) - Part 503:
CIM-XML model exchange format

SITES INTERNET

<http://cimuser.org>

- Présentations et papiers sur CIM
- Présentations des recommandations relatives à CIM
- Modèle CIM avec la représentation UML

<http://www.langdale.com.au>

- Présentations sur le modèle CIM
- Outils de conversion téléchargeables

<http://www.msdn.com>

- Informations sur l'environnement .NET et ses fonctionnalités
- Informations sur la serialisation XML
- Outils du framework .NET

<http://journaldunet.com>

- Informations sur les services Web

<http://www.ToutJavaScript.com>

- Documentation sur JavaScript

ANNEXES

Annexe A : Description des tests d'interopérabilité

Annexe B : Réseau test à 6 barres : description, fichier CIM-XML et résultats de *load flow*

Annexe C : Fichiers résultats de la conversion d'un fichier XML en schéma XML

Annexe D : Netlist du réseau à 6 barres importé

Annexe E : Netlist du réseau à 60 barres importé

Annexe F : Résultat de load flow du réseau à 6 barres importé

Annexe G : Circuit du réseau à 60 barres importé dans EMTP-RV

Annexe H : L'annuaire UDDI

Annexe I : Méthodes et propriétés pour l'objet XMLHttpRequest

Annexe J : Résumé des espaces de noms utilisés

Annexe L : Liste des fichiers du CD-ROM

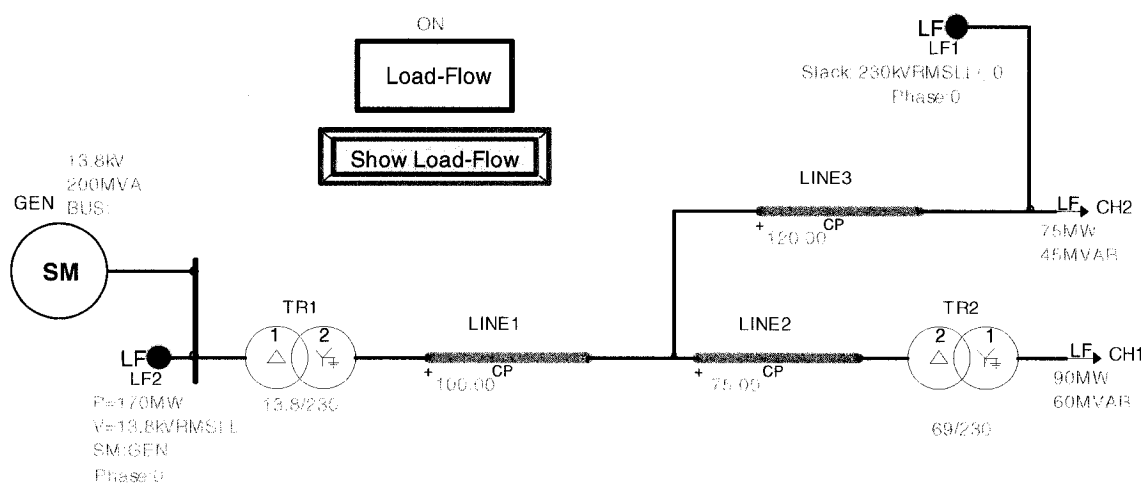
ANNEXE A : Description des tests d'interopérabilité

Test #	Date et lieu	Participants (produit)	CIM Version	Tests
1	Décembre 2000, Orlando, Florida	ABB (RDE Data Engineering Tool) – Alstom ESCA (eTerra Modeler) – PsyCor (ODMS) – Siemens (Engineering System) – SISCO (UIB Utility Integration Bus) – CIM-Logic (JCIM) – Langdale Consultants (I-Builder)	CIM u09a	- tests d'importation et d'exportation de modèles de réseaux basés sur le standard CIM/XML - interopérabilité entre deux produits : capacité d'un produit à importer correctement le modèle initialement exporté par un produit
2	Avril 2001, Las Vegas	ABB (RDE Data Engineering Tool) – Alstom ESCA (eTerra Modeler, Study Power Flow) – Siemens (IMM Information Model Manager), SISCO (UIB Utility Integration Bus), CIM-Logic (JCIM)	CIM u09b	- tests d'importation et d'exportation de modèles de réseaux basés sur le standard CIM/XML - interopérabilité entre deux produits : capacité d'un produit à importer correctement le modèle initialement exporté par un produit
3	Septembre 2001, Monterey, Californie	ABB (SABLE) – Alstom ESCA (eTerra Modeler, Study Power Flow) – Siemens (IMM Information Model Manager) – SISCO (UIB Utility Integration Bus) – PsyCor (ODMS)	CIM 10	- exécution d'applications d'écoulement de puissance (load flow) pour vérifier que le contenu des modèles échangés est correct

Test #	Date et lieu	Participants (produit)	Version CIM	Tests
4	Juillet 2002, San Francisco, Californie	ABB (SABLE) – GE Network Solutions (XA21) Langdale Consultants (CIMBuilder) – PTI (ODMS)	CIM 10	- tests d'importation et d'exportation de modèles de réseaux basés sur le standard CIM/XML - interopérabilité entre deux produits : capacité d'un produit à importer correctement le modèle initialement exporté par un produit - test de la capacité d'un produit à importer un modèle partiel de réseau et de l'ajouter à un modèle existant - exécution d'applications d'écoulement de puissance (load flow) pour vérifier que le contenu des modèles échangés est correct
5	Novembre 2003, Cleveland, Ohio	Alstom ESCA (e-Terra Modeler, Study Power Flow) – Shaw PTI (ODMS) – SISCO (UIB) – SNC Lavalin (GEN4 EMS)		
6	Juillet 2004, Folsom, Californie	Areva (e-Terra Modeler) – EDF (PRAO, EDF CIM Framework, ...) – Incremental Systems/Power Data (OTS Operator training Simulator) – Shaw PTI (ODMS) – Siemens (Spectrum PowerCC) – SISCO (UIB)		
7	Septembre 2005, Folsom, Californie	ABB, Areva, EDF, Siemens PTI, SISCO		
8	Mars 2006, San Francisco, Californie	ABB, Areva, Siemens PTI, SISCO, SNC Lavalin ECS		

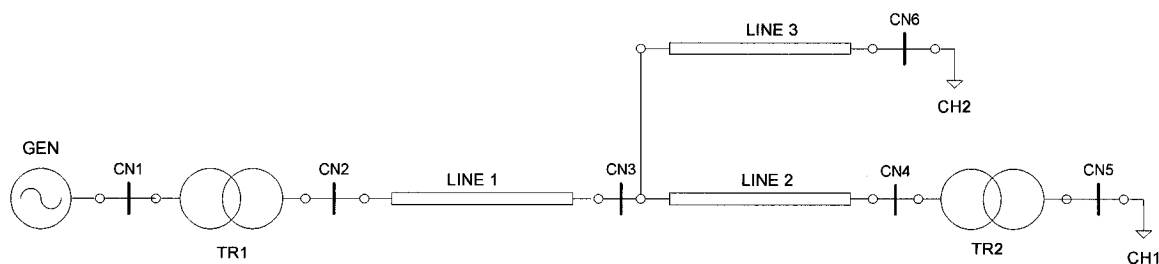
ANNEXE B : Réseau test à 6 barres (res_PM.xml)

Le réseau test à 6 barres est un réseau fictif comportant 1 machine synchrone GEN, 2 charges CH1 et CH2, deux transformateurs TR1 et TR2, 3 lignes (LINE 1, LINE2 et LINE3), comme le montre le schéma ci-dessus.



Nous avons rajouté les éléments pour effectuer un écoulement de puissance. Les résultats du load flow sont joints à la fin de cette annexe.

Nous représentons également un schéma pour sa traduction sous format CIM-XML, schéma dans lequel on met en évidence les "Terminal" de chaque élément ainsi que les "ConnectivityNode" (CN).



Le fichier CIM-XML (res_PM.xml) écrit pour ce petit réseau est le suivant :

Fichier res_PM.xml

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:cim="http://iec.ch/TC57/2003/CIM-schema-cim10#">
  <cim:ACLineSegment rdf:ID="LINE1" >
    <cim:Conductor.b0ch>0.0000029669</cim:Conductor.b0ch>
    <cim:Conductor.bch>0.0000047501</cim:Conductor.bch>
    <cim:Conductor.g0ch>0</cim:Conductor.g0ch>
    <cim:Conductor.gch>0</cim:Conductor.gch>
    <cim:Conductor.r>0.0243404</cim:Conductor.r>
    <cim:Conductor.r0>0.316764</cim:Conductor.r0>
    <cim:Conductor.x>0.3482644</cim:Conductor.x>
    <cim:Conductor.x0>1.2147</cim:Conductor.x0>
    <cim:Conductor.length>100</cim:Conductor.length>
    <cim:ConductingEquipment.BaseVoltage rdf:resource="#VOLT2"/>
    <cim:ConductingEquipment.Terminals rdf:resource="#T1_LINE1"/>
    <cim:ConductingEquipment.Terminals rdf:resource="#T2_LINE1"/>
  </cim:ACLineSegment>
  <cim:ACLineSegment rdf:ID="LINE2" >
    <cim:Conductor.b0ch>0.0000029669</cim:Conductor.b0ch>
    <cim:Conductor.bch>0.0000047501</cim:Conductor.bch>
    <cim:Conductor.g0ch>0</cim:Conductor.g0ch>
    <cim:Conductor.gch>0</cim:Conductor.gch>
    <cim:Conductor.r>0.0243404</cim:Conductor.r>
    <cim:Conductor.r0>0.316764</cim:Conductor.r0>
    <cim:Conductor.x>0.3482644</cim:Conductor.x>
    <cim:Conductor.x0>1.2147</cim:Conductor.x0>
    <cim:Conductor.length>75</cim:Conductor.length>
    <cim:ConductingEquipment.BaseVoltage rdf:resource="#VOLT2"/>
    <cim:ConductingEquipment.Terminals rdf:resource="#T1_LINE2"/>
    <cim:ConductingEquipment.Terminals rdf:resource="#T2_LINE2"/>
  </cim:ACLineSegment>
  <cim:ACLineSegment rdf:ID="LINE3" >
    <cim:Conductor.b0ch>0.0000029669</cim:Conductor.b0ch>
    <cim:Conductor.bch>0.0000047501</cim:Conductor.bch>
    <cim:Conductor.g0ch>0</cim:Conductor.g0ch>
    <cim:Conductor.gch>0</cim:Conductor.gch>
    <cim:Conductor.r>0.0243404</cim:Conductor.r>
    <cim:Conductor.r0>0.316764</cim:Conductor.r0>
    <cim:Conductor.x>0.3482644</cim:Conductor.x>
  </cim:ACLineSegment>
```

```

<cim:Conductor.x0>1.2147</cim:Conductor.x0>
<cim:Conductor.length>120</cim:Conductor.length>
<cim:ConductingEquipment.BaseVoltage rdf:resource="#VOLT3"/>
<cim:ConductingEquipment.Terminals rdf:resource="#T1_LINE3"/>
<cim:ConductingEquipment.Terminals rdf:resource="#T2_LINE3"/>
</cim:ACLineSegment>
<cim:BaseVoltage rdf:ID="VOLT1" >
  <cim:BaseVoltage.nominalVoltage>13.8</cim:BaseVoltage.nominalVoltage>
  <cim:BaseVoltage.VoltageLevel rdf:resource="#VL1"/>
</cim:BaseVoltage>
<cim:BaseVoltage rdf:ID="VOLT2" >
  <cim:BaseVoltage.nominalVoltage>230</cim:BaseVoltage.nominalVoltage>
  <cim:BaseVoltage.ConductingEquipment rdf:resource="#LINE1"/>
  <cim:BaseVoltage.ConductingEquipment rdf:resource="#LINE2"/>
  <cim:BaseVoltage.VoltageLevel rdf:resource="#VL2"/>
</cim:BaseVoltage>
<cim:BaseVoltage rdf:ID="VOLT3" >
  <cim:BaseVoltage.nominalVoltage>69</cim:BaseVoltage.nominalVoltage>
  <cim:BaseVoltage.ConductingEquipment rdf:resource="#LINE3"/>
  <cim:BaseVoltage.VoltageLevel rdf:resource="#VL3"/>
</cim:BaseVoltage>
<cim:ConnectivityNode rdf:ID="CN1">
  <cim:ConnectivityNode.MemberOf_EquipmentContainer rdf:resource="#VL1"/>
  <cim:ConnectivityNode.Terminals rdf:resource="#T_GEN" />
  <cim:ConnectivityNode.Terminals rdf:resource="#T1_TR1" />
</cim:ConnectivityNode>
<cim:ConnectivityNode rdf:ID="CN2">
  <cim:ConnectivityNode.MemberOf_EquipmentContainer rdf:resource="#VL2"/>
  <cim:ConnectivityNode.Terminals rdf:resource="#T2_TR1" />
  <cim:ConnectivityNode.Terminals rdf:resource="#T1_LINE1" />
</cim:ConnectivityNode>
<cim:ConnectivityNode rdf:ID="CN3">
  <cim:ConnectivityNode.MemberOf_EquipmentContainer rdf:resource="#VL2"/>
  <cim:ConnectivityNode.Terminals rdf:resource="#T2_LINE1" />
  <cim:ConnectivityNode.Terminals rdf:resource="#T1_LINE2" />
  <cim:ConnectivityNode.Terminals rdf:resource="#T1_LINE3" />
</cim:ConnectivityNode>
<cim:ConnectivityNode rdf:ID="CN4">
  <cim:ConnectivityNode.MemberOf_EquipmentContainer rdf:resource="#VL2"/>
  <cim:ConnectivityNode.Terminals rdf:resource="#T1_TR2" />
  <cim:ConnectivityNode.Terminals rdf:resource="#T2_LINE2" />

```

```

</cim:ConnectivityNode>
<cim:ConnectivityNode rdf:ID="CN5">
  <cim:ConnectivityNode.MemberOf_EquipmentContainer rdf:resource="#VL3"/>
  <cim:ConnectivityNode.Terminals rdf:resource="#T2_TR2"/>
  <cim:ConnectivityNode.Terminals rdf:resource="#T_CH1"/>
</cim:ConnectivityNode>
<cim:ConnectivityNode rdf:ID="CN6">
  <cim:ConnectivityNode.MemberOf_EquipmentContainer rdf:resource="#VL2"/>
  <cim:ConnectivityNode.Terminals rdf:resource="#T_CH2"/>
  <cim:ConnectivityNode.Terminals rdf:resource="#T2_LINE3"/>
</cim:ConnectivityNode>
<cim:EnergyConsumer rdf:ID="CH1">
  <cim:ConductingEquipment.Terminals rdf:resource="#T_CH1"/>
  <cim:Equipment.MemberOf_EquipmentContainer rdf:resource="#VL3"/>
  <cim:EnergyConsumer.pnom>90</cim:EnergyConsumer.pnom>
  <cim:EnergyConsumer.qnom>60</cim:EnergyConsumer.qnom>
</cim:EnergyConsumer>
<cim:EnergyConsumer rdf:ID="CH2">
  <cim:ConductingEquipment.Terminals rdf:resource="#T_CH2"/>
  <cim:Equipment.MemberOf_EquipmentContainer rdf:resource="#VL2"/>
  <cim:EnergyConsumer.pnom>75</cim:EnergyConsumer.pnom>
  <cim:EnergyConsumer.qnom>45</cim:EnergyConsumer.qnom>
</cim:EnergyConsumer>
<cim:PowerTransformer rdf:ID="TR1">
  <cim:PowerTransformer.Contains_TransformerWindings rdf:resource="#TR1_ENR1"/>
  <cim:PowerTransformer.Contains_TransformerWindings rdf:resource="#TR1_ENR2"/>
</cim:PowerTransformer>
<cim:PowerTransformer rdf:ID="TR2">
  <cim:PowerTransformer.Contains_TransformerWindings rdf:resource="#TR2_ENR1"/>
  <cim:PowerTransformer.Contains_TransformerWindings rdf:resource="#TR2_ENR2"/>
</cim:PowerTransformer>
<cim:SynchronousMachine rdf:ID="GEN">
  <cim:SynchronousMachine.minimumMVA>-1000</cim:SynchronousMachine.minimumMVA>
  <cim:SynchronousMachine.maximumMVA>1000</cim:SynchronousMachine.maximumMVA>
  <cim:SynchronousMachine.ratedMVA>200</cim:SynchronousMachine.ratedMVA>
  <cim:SynchronousMachine.damping>0</cim:SynchronousMachine.damping>
  <cim:SynchronousMachine.inertia>4.5</cim:SynchronousMachine.inertia>
  <cim:SynchronousMachine.operatingMode rdf:resource="http://iec.ch/TC57/2003/CIM-schema-cim10#SynchronousMachineOperatingMode.generator"/>
  <cim:SynchronousMachine.type rdf:resource="http://iec.ch/TC57/2003/CIM-schema-cim10#SynchronousMachineType.generator"/>

```

```

<cim:SynchronousMachine.r>0.002</cim:SynchronousMachine.r>
<cim:SynchronousMachine.x>0.14</cim:SynchronousMachine.x>
<cim:SynchronousMachine.x0>0.1</cim:SynchronousMachine.x0>
<cim:SynchronousMachine.xDirectSync>1.0</cim:SynchronousMachine.xDirectSync>
<cim:SynchronousMachine.xDirectTrans>0.000742</cim:SynchronousMachine.xDirectTrans>
<cim:SynchronousMachine.xQuadSync>1.55</cim:SynchronousMachine.xQuadSync>
<cim:SynchronousMachine.xDirectTrans>0.4</cim:SynchronousMachine.xDirectTrans>
<cim:SynchronousMachine.xDirectSubtrans>0</cim:SynchronousMachine.xDirectSubtrans>
<cim:Equipment.MemberOf_EquipmentContainer rdf:resource="#VL1"/>
<cim:ConductingEquipment.Terminals rdf:resource="#T_GEN"/>
</cim:SynchronousMachine>
<cim:Terminal rdf:ID="T_GEN">
<cim:Terminal.ConductingEquipment rdf:resource="#GEN" />
<cim:Terminal.ConnectivityNode rdf:resource="#CN1" />
</cim:Terminal>
<cim:Terminal rdf:ID="T1_TR1">
<cim:Terminal.ConductingEquipment rdf:resource="#TR1_ENR1" />
<cim:Terminal.ConnectivityNode rdf:resource="#CN1" />
</cim:Terminal>
<cim:Terminal rdf:ID="T2_TR1">
<cim:Terminal.ConductingEquipment rdf:resource="#TR1_ENR2" />
<cim:Terminal.ConnectivityNode rdf:resource="#CN2" />
</cim:Terminal>
<cim:Terminal rdf:ID="T1_LINE1">
<cim:Terminal.ConductingEquipment rdf:resource="#LINE1" />
<cim:Terminal.ConnectivityNode rdf:resource="#CN2" />
</cim:Terminal>
<cim:Terminal rdf:ID="T2_LINE1">
<cim:Terminal.ConductingEquipment rdf:resource="#LINE1" />
<cim:Terminal.ConnectivityNode rdf:resource="#CN3" />
</cim:Terminal>
<cim:Terminal rdf:ID="T1_LINE2">
<cim:Terminal.ConductingEquipment rdf:resource="#LINE2" />
<cim:Terminal.ConnectivityNode rdf:resource="#CN3" />
</cim:Terminal>
<cim:Terminal rdf:ID="T2_LINE2">
<cim:Terminal.ConductingEquipment rdf:resource="#LINE2" />
<cim:Terminal.ConnectivityNode rdf:resource="#CN4" />
</cim:Terminal>
<cim:Terminal rdf:ID="T1_TR2">
<cim:Terminal.ConductingEquipment rdf:resource="#TR2_ENR1" />

```

```

    <cim:Terminal.ConnectivityNode rdf:resource="#CN4" />
  </cim:Terminal>
<cim:Terminal rdf:ID="T2_TR2">
  <cim:Terminal.ConductingEquipment rdf:resource="#TR2_ENR2" />
  <cim:Terminal.ConnectivityNode rdf:resource="#CN5" />
</cim:Terminal>
<cim:Terminal rdf:ID="T_CH1">
  <cim:Terminal.ConductingEquipment rdf:resource="#CH1" />
  <cim:Terminal.ConnectivityNode rdf:resource="#CN5" />
</cim:Terminal>
<cim:Terminal rdf:ID="T1_LINE3">
  <cim:Terminal.ConductingEquipment rdf:resource="#LINE3" />
  <cim:Terminal.ConnectivityNode rdf:resource="#CN3" />
</cim:Terminal>
<cim:Terminal rdf:ID="T2_LINE3">
  <cim:Terminal.ConductingEquipment rdf:resource="#LINE3" />
  <cim:Terminal.ConnectivityNode rdf:resource="#CN6" />
</cim:Terminal>
<cim:Terminal rdf:ID="T_CH2">
  <cim:Terminal.ConductingEquipment rdf:resource="#CH2" />
  <cim:Terminal.ConnectivityNode rdf:resource="#CN6" />
</cim:Terminal>
<cim:TransformerWinding rdf:ID="TR1_ENR1" >
  <cim:TransformerWinding.b>0</cim:TransformerWinding.b>
  <cim:TransformerWinding.g>0</cim:TransformerWinding.g>
  <cim:TransformerWinding.r>0.0015</cim:TransformerWinding.r>
  <cim:TransformerWinding.ratedKV>13.8</cim:TransformerWinding.ratedKV>
  <cim:TransformerWinding.ratedMVA>200</cim:TransformerWinding.ratedMVA>
  <cim:TransformerWinding.windingType rdf:resource="http://iec.ch/TC57/2003/CIM-schema-cim10#WindingType.primary"/>
  <cim:TransformerWinding.x>0.1</cim:TransformerWinding.x>
  <cim:TransformerWinding.connectionType rdf:resource="http://iec.ch/TC57/2003/CIM-schema-cim10#WindingConnection.D"/>
  <cim:TransformerWinding.grounded>false</cim:TransformerWinding.grounded>
  <cim:TransformerWinding.MemberOf_PowerTransformer rdf:resource="#TR1"/>
  <cim:Equipment.MemberOf_EquipmentContainer rdf:resource="#VL1"/>
  <cim:ConductingEquipment.Terminals rdf:resource="#T1_TR1"/>
</cim:TransformerWinding>
<cim:TransformerWinding rdf:ID="TR1_ENR2" >
  <cim:TransformerWinding.b>0</cim:TransformerWinding.b>
  <cim:TransformerWinding.g>0</cim:TransformerWinding.g>

```

```

<cim:TransformerWinding.r>0</cim:TransformerWinding.r>
<cim:TransformerWinding.ratedKV>230</cim:TransformerWinding.ratedKV>
<cim:TransformerWinding.ratedMVA>200</cim:TransformerWinding.ratedMVA>
<cim:TransformerWinding.windingType rdf:resource="http://iec.ch/TC57/2003/CIM-schema-
cim10#windingType.secondary"/>
<cim:TransformerWinding.x>0</cim:TransformerWinding.x>
<cim:TransformerWinding.connectionType rdf:resource="http://iec.ch/TC57/2003/CIM-schema-
cim10#windingConnection.Y"/>
<cim:TransformerWinding.grounded>true</cim:TransformerWinding.grounded>
<cim:TransformerWinding.MemberOf_PowerTransformer rdf:resource="#TR1"/>
<cim:Equipment.MemberOf_EquipmentContainer rdf:resource="#VL2"/>
<cim:ConductingEquipment.Terminals rdf:resource="#T2_TR1"/>
</cim:TransformerWinding>
<cim:TransformerWinding rdf:ID="TR2_ENR1" >
  <cim:TransformerWinding.b>0</cim:TransformerWinding.b>
  <cim:TransformerWinding.g>0</cim:TransformerWinding.g>
  <cim:TransformerWinding.r>0.0025</cim:TransformerWinding.r>
  <cim:TransformerWinding.ratedKV>230</cim:TransformerWinding.ratedKV>
  <cim:TransformerWinding.ratedMVA>200</cim:TransformerWinding.ratedMVA>
  <cim:TransformerWinding.windingType rdf:resource="http://iec.ch/TC57/2003/CIM-schema-
cim10#windingType.primary"/>
  <cim:TransformerWinding.x>0.1</cim:TransformerWinding.x>
  <cim:TransformerWinding.connectionType rdf:resource="http://iec.ch/TC57/2003/CIM-schema-
cim10#windingConnection.Y"/>
  <cim:TransformerWinding.grounded>true</cim:TransformerWinding.grounded>
  <cim:TransformerWinding.MemberOf_PowerTransformer rdf:resource="#TR2"/>
  <cim:Equipment.MemberOf_EquipmentContainer rdf:resource="#VL2"/>
  <cim:ConductingEquipment.Terminals rdf:resource="#T1_TR2"/>
</cim:TransformerWinding>
<cim:TransformerWinding rdf:ID="TR2_ENR2" >
  <cim:TransformerWinding.b>0</cim:TransformerWinding.b>
  <cim:TransformerWinding.g>0</cim:TransformerWinding.g>
  <cim:TransformerWinding.r>0</cim:TransformerWinding.r>
  <cim:TransformerWinding.ratedKV>69</cim:TransformerWinding.ratedKV>
  <cim:TransformerWinding.ratedMVA>200</cim:TransformerWinding.ratedMVA>
  <cim:TransformerWinding.windingType rdf:resource="http://iec.ch/TC57/2003/CIM-schema-
cim10#windingType.secondary"/>
  <cim:TransformerWinding.x>0</cim:TransformerWinding.x>
  <cim:TransformerWinding.connectionType rdf:resource="http://iec.ch/TC57/2003/CIM-schema-
cim10#windingConnection.D"/>
  <cim:TransformerWinding.grounded>false</cim:TransformerWinding.grounded>

```

```

<cim:TransformerWinding.MemberOf_PowerTransformer rdf:resource="#TR2"/>
<cim:Equipment.MemberOf_EquipmentContainer rdf:resource="#VL3"/>
<cim:ConductingEquipment.Terminals rdf:resource="#T2_TR2"/>
</cim:TransformerWinding>
<cim:VoltageLevel rdf:ID="VL1" >
  <cim:VoltageLevel.BaseVoltage rdf:resource="#VOLT1"/>
  <cim:EquipmentContainer.ConnectivityNodes rdf:resource="#CN1"/>
  <cim:EquipmentContainer.Contains_Equipments rdf:resource="#GEN"/>
  <cim:EquipmentContainer.Contains_Equipments rdf:resource="#TR1_ENR1"/>
</cim:VoltageLevel>
<cim:VoltageLevel rdf:ID="VL2" >
  <cim:VoltageLevel.BaseVoltage rdf:resource="#VOLT2"/>
  <cim:EquipmentContainer.ConnectivityNodes rdf:resource="#CN2"/>
  <cim:EquipmentContainer.ConnectivityNodes rdf:resource="#CN3"/>
  <cim:EquipmentContainer.ConnectivityNodes rdf:resource="#CN4"/>
  <cim:EquipmentContainer.ConnectivityNodes rdf:resource="#CN6"/>
  <cim:EquipmentContainer.ConnectivityNodes rdf:resource="#CH2"/>
  <cim:EquipmentContainer.Contains_Equipments rdf:resource="#TR1_ENR2"/>
  <cim:EquipmentContainer.Contains_Equipments rdf:resource="#TR2_ENR1"/>
</cim:VoltageLevel>
<cim:VoltageLevel rdf:ID="VL3" >
  <cim:VoltageLevel.BaseVoltage rdf:resource="#VOLT3"/>
  <cim:EquipmentContainer.ConnectivityNodes rdf:resource="#CN5"/>
  <cim:EquipmentContainer.Contains_Equipments rdf:resource="#CH1"/>
  <cim:EquipmentContainer.Contains_Equipments rdf:resource="#TR2_ENR2"/>
</cim:VoltageLevel>
</rdf:RDF>

```

Résultat de load flow du réseau à 6 barres

EMTP output, Load-Flow solution, test case: D:\JavascriptFiles\res_P\linit - Windows Internet Explorer

Live Search

EMTP output, Load-Flow solution, test case: D:\JavascriptFiles\res_P\linit

EMTP Load-Flow Solution

- Load-Flow for Design file: D:\JavascriptFiles\res_P\linit

Solution date: 07/04/10
Solution time: 15:33:22.765

Device	Type	Vabc (kV RMSLL, deg)	phasor	P (W)	Q (VAR)	Eabc (kV RMSLL, deg)	phasor	Iabc (A, deg)	phasor
LF2	Slack	+0.2300000000E+03	+0.6788838829E-15	+0.2757586405E+07	+0.2353231529E+08	+0.2300315757E+03	-0.9413031643E-03	+0.8411099283E+02	-0.9888823183E+02
		+0.2300000000E+03	-0.1200000000E+03			+0.2300315757E+03	-0.1200000000E+03	+0.8411099283E+02	+0.1433162684E+03
		+0.2300000000E+03	+0.1200000000E+03			+0.2300315757E+03	+0.1199999987E+03	+0.8411099283E+02	+0.2331626837E+02
LF1	PVbus	+0.1380000000E+02	-0.1538670811E+02	+0.1700000000E+09	-0.7992846086E+07	+0.1641149751E+02	+0.2001827318E+02	+0.1006940182E+05	-0.1269486868E+02
		+0.1380000000E+02	-0.1353867081E+03			+0.1641149751E+02	-0.9998072682E+02	+0.1006940182E+05	-0.1326948969E+03
		+0.1380000000E+02	+0.1046132919E+03			+0.1641149751E+02	+0.1400192732E+03	+0.1006940182E+05	+0.1073051031E+03
Total Generation									
CH1a	PQload	+0.6953144024E+02	-0.3158910229E+02	+0.1672424036E+09	+0.1553966920E+08			+0.1056855604E+04	-0.3158914048E+02
CH1b	PQload	+0.6953144024E+02	-0.1515891023E+03	+0.3000000000E+08	+0.2000000000E+02			+0.1056855604E+04	-0.1515891406E+03
CH1c	PQload	+0.6953144024E+02	+0.8841089771E+02	+0.3000000000E+08	+0.2000000000E+02			+0.1056855604E+04	+0.8841085592E+02
CH2a	PQload	+0.2300000000E+03	-0.6788838829E-15	+0.2500000000E+08	+0.1500000000E+08			+0.3104968882E+03	-0.3099375653E+02
CH2b	PQload	+0.2300000000E+03	-0.1200000000E+03	+0.2500000000E+08	+0.1500000000E+08			+0.3104968882E+03	-0.1509637565E+03
CH2c	PQload	+0.2300000000E+03	+0.1200000000E+03	+0.2500000000E+08	+0.1500000000E+08			+0.3104968882E+03	+0.8903624347E+02
Total Loads									
Total Generation-Loads									
Solution frequency: 60									

Done

My Computer 75%

ANNEXE C : Fichiers résultats de la conversion d'un fichier XML en schéma XML

Le fichier XML utilisé est celui de la figure 1.8, que l'on nomme **cim_doc.xml** :

En utilisant l'outil *xsd.exe*, les fichiers résultats suivants sont obtenus :

- **cim_doc.xsd** pour l'espace de nom RDF
- **cim_xsd_appl.xsd** pour l'espace de noms CIM

Fichier cim_doc.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="RDF" targetNamespace="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" xmlns:mstns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
microsoft-com:xml-msdata" attributeFormDefault="qualified"
elementFormDefault="qualified" xmlns:appl="http://iec.ch/TC57/2000/CIM-schema-
cim10#">
  <xs:import namespace="http://iec.ch/TC57/2000/CIM-schema-cim10#"
schemaLocation="xsdtest_appl.xsd" />
  <xs:attribute name="resource" msdata:Prefix="rdf" type="xs:string" />
  <xs:attribute name="ID" msdata:Prefix="rdf" type="xs:string" />
  <xs:element name="RDF" msdata:IsDataSet="true" msdata:Locale="en-US"
msdata:Prefix="rdf">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="appl:ConductingEquipment.Terminals" />
        <xs:element ref="appl:Company" />
        <xs:element ref="appl:Bay" />
        <xs:element ref="appl:Breaker" />
        <xs:element ref="appl:CurrentRelay" />
        <xs:element ref="appl:Terminal" />
        <xs:element ref="appl:ConnectivityNode" />
        <xs:element ref="appl:TransformerWinding" />
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Fichier cim_doc_appl.xsd

```
<?xml version="1.0" standalone="yes"?>
<xs:schema targetNamespace="http://iec.ch/TC57/2000/CIM-schema-cim10#"
xmlns:mstns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns="http://iec.ch/TC57/2000/CIM-schema-cim10#"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
microsoft-com:xml-msdata" attributeFormDefault="qualified"
elementFormDefault="qualified" xmlns:appl="http://iec.ch/TC57/2000/CIM-schema-
cim10#">
  <xs:import namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
schemaLocation="xsdtest.xsd" />
```

```

<xs:element name="ConductingEquipment.Terminals" msdata:Prefix="cim">
  <xs:complexType>
    <xs:attribute ref="mstns:resource" />
  </xs:complexType>
</xs:element>
<xs:element name="Company" msdata:Prefix="cim">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Naming.name" msdata:Prefix="cim" type="xs:string"
minOccurs="0" msdata:Ordinal="0" />
    </xs:sequence>
    <xs:attribute ref="mstns:ID" />
  </xs:complexType>
</xs:element>
<xs:element name="Bay" msdata:Prefix="cim">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Naming.name" msdata:Prefix="cim" type="xs:string"
minOccurs="0" msdata:Ordinal="0" />
      <xs:element name="PowerSystemResource.MemberOfCompany"
msdata:Prefix="cim" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:attribute ref="mstns:resource" />
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute ref="mstns:ID" />
  </xs:complexType>
</xs:element>
<xs:element name="Breaker" msdata:Prefix="cim">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Naming.name" msdata:Prefix="cim" type="xs:string"
minOccurs="0" msdata:Ordinal="0" />
      <xs:element name="PowerSystemResource.ChildOf" msdata:Prefix="cim"
minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:attribute ref="mstns:resource" />
        </xs:complexType>
      </xs:element>
      <xs:element ref="ConductingEquipment.Terminals" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute ref="mstns:ID" />
  </xs:complexType>
</xs:element>
<xs:element name="CurrentRelay" msdata:Prefix="cim">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Naming.name" msdata:Prefix="cim" type="xs:string"
minOccurs="0" msdata:Ordinal="0" />
      <xs:element name="ProtectionEquipment.Operates" msdata:Prefix="cim"
minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:attribute ref="mstns:resource" />
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute ref="mstns:ID" />
  </xs:complexType>
</xs:element>

```

```

    </xs:complexType>
  </xs:element>
  <xs:element name="Terminal" msdata:Prefix="cim">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Naming.name" msdata:Prefix="cim" type="xs:string"
minOccurs="0" msdata:Ordinal="0" />
        <xs:element name="Terminal.ConductingEquipment" msdata:Prefix="cim"
minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute ref="mstns:resource" />
          </xs:complexType>
        </xs:element>
        <xs:element name="Terminal.ConnectivityNode" msdata:Prefix="cim"
minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute ref="mstns:resource" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute ref="mstns:ID" />
    </xs:complexType>
  </xs:element>
  <xs:element name="ConnectivityNode" msdata:Prefix="cim">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Naming.name" msdata:Prefix="cim" type="xs:string"
minOccurs="0" msdata:Ordinal="0" />
        <xs:element name="ConnectivityNode.Terminals" msdata:Prefix="cim"
minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute ref="mstns:resource" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute ref="mstns:ID" />
    </xs:complexType>
  </xs:element>
  <xs:element name="TransformerWinding" msdata:Prefix="cim">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Naming.name" msdata:Prefix="cim" type="xs:string"
minOccurs="0" msdata:Ordinal="0" />
        <xs:element ref="ConductingEquipment.Terminals" minOccurs="0"
maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute ref="mstns:ID" />
    </xs:complexType>
  </xs:element>
</xs:schema>

```



```

69kVRMSLL,90MW 0 60MVA 0 60 1 0
_PQload; CH2a;1;1; CN6a,
230kVRMSLL,75MW 0 45MVA 0 60 1 0
_PQload; CH2b;1;1; CN6b,
230kVRMSLL,75MW 0 45MVA 0 60 1 0
_PQload; CH2c;1;1; CN6c,
230kVRMSLL,75MW 0 45MVA 0 60 1 0
!
! Subcircuit definitions
!
<xfmr_DYgp30_unit_ecf686ce;4;i,j,k,m,
_RLC;RL1;2;2;i,s36,
#R1#,#L1#,0,0,0,
#W1_scope#,
_RLC;RL2;2;2;s31,k,
#R2#,#L2#,0,0,0,
#W2_scope#,
_Tr0;Tygyg;4;4;s36,j,s31,m
#Ratio#,,,
_x_Lnonl;Lmag;2;2;s36,j,
1,#Phi0#,1e-08,
#Lmag_scope#,
#ILnonl# #PhiLnonl#
_x_Rmag;Rmag;2;2;s36,j,
#Rm#,,,,,
>
<DYp30_O_b34e1b4e;6;DELTAA,DELTAAb,DELTAc,Ya,Yb,Yc,
@xfmr_DYgp30_unit_ecf686ce;xfmr_A;4;DELTAA,DELTAAb,Ya,,
Phi0=Phiss01
@xfmr_DYgp30_unit_ecf686ce;xfmr_B;4;DELTAAb,DELTAc,Yb,,
Phi0=Phiss02
@xfmr_DYgp30_unit_ecf686ce;xfmr_C;4;DELTAc,DELTAA,Yc,,
Phi0=Phiss03
>
<xfmr_YgDp30_unit_ed9c4fd7;4;i,j,k,m,
_RLC;RL1;2;2;i,s36,
#R1#,#L1#,0,0,0,
#W1_scope#,
_RLC;RL2;2;2;s31,k,
#R2#,#L2#,0,0,0,
#W2_scope#,
_Tr0;Tygyg;4;4;s36,j,s31,m
#Ratio#,,,
_x_Lnonl;Lmag;2;2;s36,j,
1,#Phi0#,1e-08,
#Lmag_scope#,
#ILnonl# #PhiLnonl#
_x_Rmag;Rmag;2;2;s36,j,
#Rm#,,,,,
>
<YgDp30_O_b34630b2;6;Ya,Yb,Yc,DELTAA,DELTAAb,DELTAc,
@xfmr_YgDp30_unit_ed9c4fd7;xfmr_A;4;Ya,,DELTAA,DELTAAb,
Phi0=Phiss01
@xfmr_YgDp30_unit_ed9c4fd7;xfmr_B;4;Yb,,DELTAAb,DELTAc,
Phi0=Phiss02
@xfmr_YgDp30_unit_ed9c4fd7;xfmr_C;4;Yc,,DELTAc,DELTAA,
Phi0=Phiss03
>
!
```

```

!
@DYgp30_O_b34e1b4e;TR1;6;CN1a,CN1b,CN1c,CN2a,CN2b,CN2c,
R1=0.0015;
L1=0.1;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=16.66666666666667;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YgDp30_O_b34630b2;TR2;6;CN4a,CN4b,CN4c,CN5a,CN5b,CN5c,
R1=0.0025;
L1=0.1;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.519615242270663;
Lmag_scope='';
W1_scope='';
W2_scope='';

```

ANNEXE E : Netlist pour le réseau à 60 barres**Network_60barres.net**

```

Dt=1us;
tmax=32ms;
_TLM;34593a;6;2;183B913915598B02a,BE80427AE64E0E78a,
3,0,1,1,1,1,1mH,1uF,
34593b;6;2;183B913915598B02b,BE80427AE64E0E78b,
34593c;6;2;183B913915598B02c,BE80427AE64E0E78c,
1 5.95125 0.0902067511339648 -107.219663713171
1 0.892688 0.0315723618358547 0.236792713768911
0
0
0
_TLM;34528a;6;2;B1F2563B6AC8E20Ea,AA00BD737AE59A6Ba,
3,0,1,1,1,1,1mH,1uF,
34528b;6;2;B1F2563B6AC8E20Eb,AA00BD737AE59A6Bb,
34528c;6;2;B1F2563B6AC8E20Ec,AA00BD737AE59A6Bc,
1 14.283001 0.195748651340047 -49.0844124604964
1 2.14245 0.0685120278363872 1.1555483954458
0
0
0
_TLM;34560a;6;2;CB962D2D9F3F543Ca,FEAD364EBBABC91Ca,
3,0,1,1,1,1,1mH,1uF,
34560b;6;2;CB962D2D9F3F543Cb,FEAD364EBBABC91Cb,
34560c;6;2;CB962D2D9F3F543Cc,FEAD364EBBABC91Cc,
1 6.348 0.126289447343419 -77.552195971345
1 0.9522 0.0442013065701966 0.811409723054708
0
0
0
_TLM;34517a;6;2;FDC9D63486D85F78a,68186169E633A724a,
3,0,1,1,1,1,1mH,1uF,
34517b;6;2;FDC9D63486D85F78b,68186169E633A724b,
34517c;6;2;FDC9D63486D85F78c,68186169E633A724c,
1 18.2505 0.327450494096103 -29.7905313519004
1 2.737575 0.11460767081157 1.19974974976248
0
0
0
_TLM;34532a;6;2;13136855F5863B5Ca,01C48F6610722D1Fa,
3,0,1,1,1,1,1mH,1uF,
34532b;6;2;13136855F5863B5Cb,01C48F6610722D1Fb,
34532c;6;2;13136855F5863B5Cc,01C48F6610722D1Fc,
1 7.1415 0.0847943424584149 -111.968382800909
1 1.071225 0.0296780201257035 0.539887400656028
0
0
0
_TLM;34591a;6;2;C5EC267423E39F1Fa,D77D5958909FAB6Da,
3,0,1,1,1,1,1mH,1uF,
34591b;6;2;C5EC267423E39F1Fb,D77D5958909FAB6Db,
34591c;6;2;C5EC267423E39F1Fc,D77D5958909FAB6Dc,
1 5.95125 0.0902067511339648 -107.219663713171
1 0.892688 0.0315723618358547 0.236792713768911

```

0
0
0
_TLM;34501a;6;2;81A2B223A8DB4538a,8AC61C5805667907a,
3,0,1,1,1,1,1mH,1uF,
34501b;6;2;81A2B223A8DB4538b,8AC61C5805667907b,
34501c;6;2;81A2B223A8DB4538c,8AC61C5805667907c,
1 3.174 0.0387889005472291 -245.421896577092
1 0.4761 0.0135761155894175 0.230478246706904
0
0
0
_TLM;34568a;6;2;352EE750CC9F0E12a,22618876924EB524a,
3,0,1,1,1,1,1mH,1uF,
34568b;6;2;352EE750CC9F0E12b,22618876924EB524b,
34568c;6;2;352EE750CC9F0E12c,22618876924EB524c,
1 7.1415 0.0911088148255859 -104.875300226098
1 1.071225 0.0318880854542133 0.543044647449943
0
0
0
_TLM;34527a;6;2;EFD8A053826AB11Fa,B2659007C4867368a,
3,0,1,1,1,1,1mH,1uF,
34527b;6;2;EFD8A053826AB11Fb,B2659007C4867368b,
34527c;6;2;EFD8A053826AB11Fc,B2659007C4867368c,
1 10.315499 0.192140359437409 -50.8482368332504
1 1.547325 0.0672491280577882 0.697749209835302
0
0
0
_TLM;1215a;6;2;940789700C54B077a,51D8F76FC7FF4025a,
3,0,1,1,1,1,1mH,1uF,
1215b;6;2;940789700C54B077b,51D8F76FC7FF4025b,
1215c;6;2;940789700C54B077c,51D8F76FC7FF4025c,
1 0 0 0
1 0 0 0
0
0
0
_TLM;34515a;6;2;B878E07C3567CA74a,1CED135402763131a,
3,0,1,1,1,1,1mH,1uF,
34515b;6;2;B878E07C3567CA74b,1CED135402763131b,
34515c;6;2;B878E07C3567CA74c,1CED135402763131c,
1 1.587 0.0234537541276946 -411.747873642554
1 0.23805 0.00820881407732223 0.13576116119934
0
0
0
_TLM;T559a;6;2;7897704E05770161a,E2DAB632C25AD032a,
3,0,1,1,1,1,1mH,1uF,
T559b;6;2;7897704E05770161b,E2DAB632C25AD032b,
T559c;6;2;7897704E05770161c,E2DAB632C25AD032c,
1 6.348 0.12177911296982 -80.3186548055509
1 0.9522 0.0426226884784039 0.805095226814296
0
0
0
_TLM;MART_HUNT_1001a;6;2;B822583A830FFE22a,BCD071335241B21Ba,
3,0,1,1,1,1,1mH,1uF,

MART_HUNT_1001b;6;2;B822583A830FFE22b,BCD071335241B21Bb,
 MART_HUNT_1001c;6;2;B822583A830FFE22c,BCD071335241B21Bc,
 1 11.108999 0.132603927668337 0
 1 1.66635 0.0464113718987065 0
 0
 0
 0
 _TLM;34502a;6;2;959FAE2E97227A68a,EE685B3E0CF8B329a,
 3,0,1,1,1,1,1mH,1uF,
 34502b;6;2;959FAE2E97227A68b,EE685B3E0CF8B329b,
 34502c;6;2;959FAE2E97227A68c,EE685B3E0CF8B329c,
 1 27.772499 0.370749760847921 -25.8652122909111
 1 4.165875 0.129762412450528 2.20690814007273
 0
 0
 0
 _TLM;34513a;6;2;BFE8370B9D836E65a,BD2D2A07F2FA2923a,
 3,0,1,1,1,1,1mH,1uF,
 34513b;6;2;BFE8370B9D836E65b,BD2D2A07F2FA2923b,
 34513c;6;2;BFE8370B9D836E65c,BD2D2A07F2FA2923c,
 1 3.174 0.0414951048850041 -230.725995028904
 1 0.4761 0.0145232864444932 0.246264427624832
 0
 0
 0
 _TLM;34519a;6;2;94A6E95316AF6535a,1AC5B8204A196F07a,
 3,0,1,1,1,1,1mH,1uF,
 34519b;6;2;94A6E95316AF6535b,1AC5B8204A196F07b,
 34519c;6;2;94A6E95316AF6535c,1AC5B8204A196F07c,
 1 5.5545 0.0739695330650622 -129.617448687913
 1 0.833175 0.0258893367054009 0.416755170928118
 0
 0
 0
 _TLM;34514a;6;2;80D55556AB7EC176a,1E52D7654F9E3A6Fa,
 3,0,1,1,1,1,1mH,1uF,
 34514b;6;2;80D55556AB7EC176b,1E52D7654F9E3A6Fb,
 34514c;6;2;80D55556AB7EC176c,1E52D7654F9E3A6Fc,
 1 4.761 0.0829902097700081 -117.393764819228
 1 0.71415 0.0290465728889864 0.356767694050323
 0
 0
 0
 _TLM;T540a;6;2;F36BAE6E5DC3D36Fa,A14C0B3EA4F20A1Aa,
 3,0,1,1,1,1,1mH,1uF,
 T540b;6;2;F36BAE6E5DC3D36Fb,A14C0B3EA4F20A1Ab,
 T540c;6;2;F36BAE6E5DC3D36Fc,A14C0B3EA4F20A1Ac,
 1 55.544998 0.0775778037470406 -27.8903064368364
 1 8.33175 0.0271522311788351 0.460956496066391
 0
 0
 0
 _TLM;516517a;6;2;AA37B14CAF65CA1Ba,6BCED8475F1D0371a,
 3,0,1,1,1,1,1mH,1uF,
 516517b;6;2;AA37B14CAF65CA1Bb,6BCED8475F1D0371b,
 516517c;6;2;AA37B14CAF65CA1Bc,6BCED8475F1D0371c,
 1 0 0 0
 1 0 0 0
 0

0
0
_TLM;34504a;6;2;8D2ABB22040DD35Da,3A830F3E1EF1B848a,
3,0,1,1,1,1,1mH,1uF,
34504b;6;2;8D2ABB22040DD35Db,3A830F3E1EF1B848b,
34504c;6;2;8D2ABB22040DD35Dc,3A830F3E1EF1B848c,
1 7.935 0.225516867224583 -43.8195900867978
1 1.19025 0.0789309045896369 2.36792713768911
0
0
0
_TLM;34562a;6;2;0983D076B975E85Da,183B913915598B02a,
3,0,1,1,1,1,1mH,1uF,
34562b;6;2;0983D076B975E85Db,183B913915598B02b,
34562c;6;2;0983D076B975E85Dc,183B913915598B02c,
1 4.761 0.0865984804519865 -112.710510592048
1 0.71415 0.0303094673624205 0.584088693963313
0
0
0
_TLM;34503a;6;2;EC5A647CF0837657a,CBEA83459B962063a,
3,0,1,1,1,1,1mH,1uF,
34503b;6;2;EC5A647CF0837657b,CBEA83459B962063b,
34503c;6;2;EC5A647CF0837657c,CBEA83459B962063c,
1 3.174 0.0387889005472291 -245.421896577092
1 0.4761 0.0135761155894175 0.230478246706904
0
0
0
_TLM;148150a;6;2;6F56FC25C94B7A14a,9794B14836238E03a,
3,0,1,1,1,1,1mH,1uF,
148150b;6;2;6F56FC25C94B7A14b,9794B14836238E03b,
148150c;6;2;6F56FC25C94B7A14c,9794B14836238E03c,
1 0 0 0
1 0 0 0
0
0
0
_TLM;34523a;6;2;629E822A7403A252a,2F8C384C704A8879a,
3,0,1,1,1,1,1mH,1uF,
34523b;6;2;629E822A7403A252b,2F8C384C704A8879b,
34523c;6;2;629E822A7403A252c,2F8C384C704A8879c,
1 10.315499 0.156057679143449 -61.9695820189192
1 1.547325 0.0546201859760287 1.01663007498776
0
0
0
_TLM;34509a;6;2;46134844A6699753a,F30E3B38FAC8C151a,
3,0,1,1,1,1,1mH,1uF,
34509b;6;2;46134844A6699753b,F30E3B38FAC8C151b,
34509c;6;2;46134844A6699753c,F30E3B38FAC8C151c,
1 6.348 0.115464637950066 -84.5328867975753
1 0.9522 0.0404126231498941 0.423069635337542
0
0
0
_TLM;34522a;6;2;EC100A4F999D607Fa,E3D96616220B5469a,
3,0,1,1,1,1,1mH,1uF,
34522b;6;2;EC100A4F999D607Fb,E3D96616220B5469b,

34522c;6;2;EC100A4F999D607Fc,E3D96616220B5469c,
1 5.5545 0.0802840054322331 -120.115508984637
1 0.833175 0.0280994020339107 0.423069635337542
0
0
0
_TLM;T545a;6;2;34CFEF24AACDF931a,72D0BF5B263F8F73a,
3,0,1,1,1,1,1mH,1uF,
T545b;6;2;34CFEF24AACDF931b,72D0BF5B263F8F73b,
T545c;6;2;34CFEF24AACDF931c,72D0BF5B263F8F73c,
1 25.391998 0.306702946958766 -31.0056337020745
1 3.8088 0.107346032894489 1.61019045362859
0
0
0
_TLM;34542a;6;2;50AEC4189E40792Aa,700C7B36D0C2993Da,
3,0,1,1,1,1,1mH,1uF,
34542b;6;2;50AEC4189E40792Ab,700C7B36D0C2993Db,
34542c;6;2;50AEC4189E40792Ac,700C7B36D0C2993Dc,
1 25.391998 0.291367811149561 -32.4760860069125
1 3.8088 0.101978728729811 1.61966213830611
0
0
0
_TLM;34520a;6;2;C954B91953924742a,C4F53F25D462CA08a,
3,0,1,1,1,1,1mH,1uF,
34520b;6;2;C954B91953924742b,C4F53F25D462CA08b,
34520c;6;2;C954B91953924742c,C4F53F25D462CA08c,
1 6.348 0.116366712252017 -83.9045690483774
1 0.9522 0.040728349420835 0.43569859863996
0
0
0
_TLM;T525a;6;2;283D3A63E2B34055a,6F1DF6336D0AB057a,
3,0,1,1,1,1,1mH,1uF,
T525b;6;2;283D3A63E2B34055b,6F1DF6336D0AB057b,
T525c;6;2;283D3A63E2B34055c,6F1DF6336D0AB057c,
1 10.315499 0.136212190392568 -70.3423742802828
1 1.547325 0.0476742663721407 0.811409723054708
0
0
0
_TLM;HUNT_NANT_1001a;6;2;09DFA371E9B9821Ea,BCD071335241B21Ba,
3,0,1,1,1,1,1mH,1uF,
HUNT_NANT_1001b;6;2;09DFA371E9B9821Eb,BCD071335241B21Bb,
HUNT_NANT_1001c;6;2;09DFA371E9B9821Ec,BCD071335241B21Bc,
1 11.108999 0.136212190392568 0
1 1.66635 0.0476742663721407 0
0
0
0
_TLM;34521a;6;2;22AA490389CBA843a,C450945F45531E29a,
3,0,1,1,1,1,1mH,1uF,
34521b;6;2;22AA490389CBA843b,C450945F45531E29b,
34521c;6;2;22AA490389CBA843c,C450945F45531E29c,
1 8.728499 0.119974980281413 -80.1019802209082
1 1.309275 0.0419912438942692 0.67564853002438
0
0

```

0
_TLM;13869a;6;2;285FD8270ADF642Aa,810D3B4E4C730B2Da,
3,0,1,1,1,1,1mH,1uF,
13869b;6;2;285FD8270ADF642Ab,810D3B4E4C730B2Db,
13869c;6;2;285FD8270ADF642Ac,810D3B4E4C730B2Dc,
1 1.424491 0.00971635091894774 -22.8132488924384
1 0.213674 0.003400722025857 0.0255054151939275
0
0
0
_TLM;34506a;6;2;3C4CB519C7D48E56a,37A392019AEC8577a,
3,0,1,1,1,1,1mH,1uF,
34506b;6;2;3C4CB519C7D48E56b,37A392019AEC8577b,
34506c;6;2;3C4CB519C7D48E56c,37A392019AEC8577c,
1 5.5545 0.0739695330650622 -129.617448687913
1 0.833175 0.0258893367054009 0.438855816255469
0
0
0
_TLM;MART_NANT_1002a;6;2;09DFA371E9B9821Ea,B822583A830FFE22a,
3,0,1,1,1,1,1mH,1uF,
MART_NANT_1002b;6;2;09DFA371E9B9821Eb,B822583A830FFE22b,
MART_NANT_1002c;6;2;09DFA371E9B9821Ec,B822583A830FFE22c,
1 45.229497 0.563792191934698 0
1 6.784425 0.197327261474092 0
0
0
0
_TLM;1921a;6;2;6050856F99EC9D02a,60D79413706CE27Fa,
3,0,1,1,1,1,1mH,1uF,
1921b;6;2;6050856F99EC9D02b,60D79413706CE27Fb,
1921c;6;2;6050856F99EC9D02c,60D79413706CE27Fc,
1 0 0 0
1 0 0 0
0
0
0
_TLM;34563a;6;2;AAAA19740986F679a,B64802328E7DF765a,
3,0,1,1,1,1,1mH,1uF,
34563b;6;2;AAAA19740986F679b,B64802328E7DF765b,
34563c;6;2;AAAA19740986F679c,B64802328E7DF765c,
1 2.3805 0.0532219806225602 -184.693846042915
1 0.357075 0.0186276934831543 0.214692065788977
0
0
0
_TLM;TIE_LINE_5_6_4a;6;2;558F3365A17F1C63a,75D66212A0924429a,
3,0,1,1,1,1,1mH,1uF,
TIE_LINE_5_6_4b;6;2;558F3365A17F1C63b,75D66212A0924429b,
TIE_LINE_5_6_4c;6;2;558F3365A17F1C63c,75D66212A0924429c,
1 17.456999 0.315723618358547 -30.9075077113244
1 2.61855 0.110503266425492 1.13976221452787
0
0
0
_TLM;T538a;6;2;0E3E3D273D2E0860a,CCC832566093F215a,
3,0,1,1,1,1,1mH,1uF,
T538b;6;2;0E3E3D273D2E0860b,CCC832566093F215b,
T538c;6;2;0E3E3D273D2E0860c,CCC832566093F215c,

```

1 12.695999 0.175903165241748 -54.66465012964
1 1.9044 0.0615661082324991 0.959799786547072
0
0
0
_TLM;500502a;6;2;ED9C1C4A4EE16C27a,28E74E02E0530E12a,
3,0,1,1,1,1,1mH,1uF,
500502b;6;2;ED9C1C4A4EE16C27b,28E74E02E0530E12b,
500502c;6;2;ED9C1C4A4EE16C27c,28E74E02E0530E12c,
1 0 0 0
1 0 0 0
0
0
0
_TLM;34512a;6;2;92971A19E7FD9967a,B5E15537832C624Ca,
3,0,1,1,1,1,1mH,1uF,
34512b;6;2;92971A19E7FD9967b,B5E15537832C624Cb,
34512c;6;2;92971A19E7FD9967c,B5E15537832C624Cc,
1 6.348 0.101031560527318 -95.9970457066494
1 0.9522 0.0353610452561573 0.467270960475815
0
0
0
_TLM;34592a;6;2;10D90C4694A96418a,5B468D4607139D69a,
3,0,1,1,1,1,1mH,1uF,
34592b;6;2;10D90C4694A96418b,5B468D4607139D69b,
34592c;6;2;10D90C4694A96418c,5B468D4607139D69c,
1 5.95125 0.0902067511339648 -107.219663713171
1 0.892688 0.0315723618358547 0.236792713768911
0
0
0
_TLM;527528a;6;2;7723D33730FEC57Ba,ECC9626CDC4CA76Ca,
3,0,1,1,1,1,1mH,1uF,
527528b;6;2;7723D33730FEC57Bb,ECC9626CDC4CA76Cb,
527528c;6;2;7723D33730FEC57Bc,ECC9626CDC4CA76Cc,
1 0 0 0
1 0 0 0
0
0
0
_TLM;13822a;6;2;B560A675909BB879a,18A575649CB8C025a,
3,0,1,1,1,1,1mH,1uF,
13822b;6;2;B560A675909BB879b,18A575649CB8C025b,
13822c;6;2;B560A675909BB879c,18A575649CB8C025c,
1 1.17311 0.00800170044471198 -27.7018024017923
1 0.175967 0.002800593829358 0.021004460351641
0
0
0
_TLM;34590a;6;2;03D9E90087EB3120a,FEAD364EBBABC91Ca,
3,0,1,1,1,1,1mH,1uF,
34590b;6;2;03D9E90087EB3120b,FEAD364EBBABC91Cb,
34590c;6;2;03D9E90087EB3120c,FEAD364EBBABC91Cc,
1 5.95125 0.0902067511339648 -107.219663713171
1 0.892688 0.0315723618358547 0.236792713768911
0
0
0

```

_TLM;T511a;6;2;24A9D97415E9C166a,DB4DD01F7A57C570a,
3,0,1,1,1,1,1mH,1uF,
T511b;6;2;24A9D97415E9C166b,DB4DD01F7A57C570b,
T511c;6;2;24A9D97415E9C166c,DB4DD01F7A57C570c,
1 7.935 0.225516867224583 -43.8195900867978
1 1.19025 0.0789309045896369 3.78868342030257
0
0
0
_SW0;_58A8CE3856417A5Ca;2;2;4613006609E16B4Da,C9C8B84BBF2BC705a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_58A8CE3856417A5Cb;2;2;4613006609E16B4Db,C9C8B84BBF2BC705b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_58A8CE3856417A5Cc;2;2;4613006609E16B4Dc,C9C8B84BBF2BC705c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_537F530FBD270233a;2;2;DA9B155D9FA61123a,EB998F2991147947a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_537F530FBD270233b;2;2;DA9B155D9FA61123b,EB998F2991147947b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_537F530FBD270233c;2;2;DA9B155D9FA61123c,EB998F2991147947c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0DBD9465BE8F624Da;2;2;9E0ECD16EC5EE41Ba,CFD9E562BDC2587Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0DBD9465BE8F624Db;2;2;9E0ECD16EC5EE41Bb,CFD9E562BDC2587Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0DBD9465BE8F624Dc;2;2;9E0ECD16EC5EE41Bc,CFD9E562BDC2587Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_EF60345FED62B518a;2;2;D5596E4146DE113Fa,9277683CC735A247a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_EF60345FED62B518b;2;2;D5596E4146DE113Fb,9277683CC735A247b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_EF60345FED62B518c;2;2;D5596E4146DE113Fc,9277683CC735A247c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_486C77446237F325a;2;2;BCB0F348D164C41Ca,D9E12F0C54F08F09a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_486C77446237F325b;2;2;BCB0F348D164C41Cb,D9E12F0C54F08F09b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_486C77446237F325c;2;2;BCB0F348D164C41Cc,D9E12F0C54F08F09c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_16F33740A85E8676a;2;2;CF721A2D7629203Aa,85E55D69ABE5930Fa,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_16F33740A85E8676b;2;2;CF721A2D7629203Ab,85E55D69ABE5930Fb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_16F33740A85E8676c;2;2;CF721A2D7629203Ac,85E55D69ABE5930Fc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_6CCE7852B413BF2Aa;2;2;C005890625AA7A17a,940789700C54B077a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_6CCE7852B413BF2Ab;2;2;C005890625AA7A17b,940789700C54B077b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_6CCE7852B413BF2Ac;2;2;C005890625AA7A17c,940789700C54B077c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_98967F3CA1A60C28a;2;2;BFE8370B9D836E65a,E2B6016E76F4B965a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_98967F3CA1A60C28b;2;2;BFE8370B9D836E65b,E2B6016E76F4B965b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_98967F3CA1A60C28c;2;2;BFE8370B9D836E65c,E2B6016E76F4B965c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_815BDF120E1A6E5Fa;2;2;FEBFC76E4BB0951Aa,1435785F25869A65a,
-1,1E+15,0,0,0,,.1,.001,5,

```

_SW0;_815BDF120E1A6E5Fb;2;2;FEBFC76E4BB0951Ab,1435785F25869A65b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_815BDF120E1A6E5Fc;2;2;FEBFC76E4BB0951Ac,1435785F25869A65c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_3A8412639BCA937Ca;2;2;A3C0FB3A6003DD67a,66D858026E93426Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_3A8412639BCA937Cb;2;2;A3C0FB3A6003DD67b,66D858026E93426Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_3A8412639BCA937Cc;2;2;A3C0FB3A6003DD67c,66D858026E93426Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C1FC994C1CFFE407a;2;2;489F1F498213AD67a,1E52D7654F9E3A6Fa,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C1FC994C1CFFE407b;2;2;489F1F498213AD67b,1E52D7654F9E3A6Fb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C1FC994C1CFFE407c;2;2;489F1F498213AD67c,1E52D7654F9E3A6Fc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D8FD376463BD2307a;2;2;0CEB5532D83A4051a,8242B9323E862241a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D8FD376463BD2307b;2;2;0CEB5532D83A4051b,8242B9323E862241b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D8FD376463BD2307c;2;2;0CEB5532D83A4051c,8242B9323E862241c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9C06B16FF5D24C29a;2;2;863D8457254C3B16a,3232311F943C7523a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9C06B16FF5D24C29b;2;2;863D8457254C3B16b,3232311F943C7523b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9C06B16FF5D24C29c;2;2;863D8457254C3B16c,3232311F943C7523c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_135426636700D810a;2;2;8F261B0CD8BECD44a,C423EE218A59BC29a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_135426636700D810b;2;2;8F261B0CD8BECD44b,C423EE218A59BC29b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_135426636700D810c;2;2;8F261B0CD8BECD44c,C423EE218A59BC29c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_86ACD667508EFC78a;2;2;A28ED34FBCEA5963a,7FB36165489E8C75a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_86ACD667508EFC78b;2;2;A28ED34FBCEA5963b,7FB36165489E8C75b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_86ACD667508EFC78c;2;2;A28ED34FBCEA5963c,7FB36165489E8C75c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_D9BE4C34FF3EBB16a;2;2;B5E15537832C624Ca,6E7617670E246633a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D9BE4C34FF3EBB16b;2;2;B5E15537832C624Cb,6E7617670E246633b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D9BE4C34FF3EBB16c;2;2;B5E15537832C624Cc,6E7617670E246633c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_4C0B39072437E80Ba;2;2;AA45C33E4BB7ED70a,1448C1713BA1374Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_4C0B39072437E80Bb;2;2;AA45C33E4BB7ED70b,1448C1713BA1374Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_4C0B39072437E80Bc;2;2;AA45C33E4BB7ED70c,1448C1713BA1374Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_024DFA57F1AB063Ba;2;2;77C3AB49608B6828a,6BCED8475F1D0371a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_024DFA57F1AB063Bb;2;2;77C3AB49608B6828b,6BCED8475F1D0371b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_024DFA57F1AB063Bc;2;2;77C3AB49608B6828c,6BCED8475F1D0371c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_A385B06AAD0AB17Ea;2;2;1CED135402763131a,A9E68607328AB530a,

-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_A385B06AAD0AB17Eb;2;2;1CED135402763131b,A9E68607328AB530b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_A385B06AAD0AB17Ec;2;2;1CED135402763131c,A9E68607328AB530c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_18DA9E749C665737a;2;2;6FB3260958FBA52Da,51175F3576CE403Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_18DA9E749C665737b;2;2;6FB3260958FBA52Db,51175F3576CE403Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_18DA9E749C665737c;2;2;6FB3260958FBA52Dc,51175F3576CE403Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_155B2E6FEB05C428a;2;2;30546F43BD0F8533a,3258DC79BD22260Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_155B2E6FEB05C428b;2;2;30546F43BD0F8533b,3258DC79BD22260Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_155B2E6FEB05C428c;2;2;30546F43BD0F8533c,3258DC79BD22260Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_A937953D847B4E3Ea;2;2;96F75313285FFE2Ca,C4F53F25D462CA08a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_A937953D847B4E3Eb;2;2;96F75313285FFE2Cb,C4F53F25D462CA08b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_A937953D847B4E3Ec;2;2;96F75313285FFE2Cc,C4F53F25D462CA08c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9E7FA139FBD78001a;2;2;CA7B405350470373a,C203F07798607476a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_9E7FA139FBD78001b;2;2;CA7B405350470373b,C203F07798607476b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_9E7FA139FBD78001c;2;2;CA7B405350470373c,C203F07798607476c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_F4456E408BF15920a;2;2;EC100A4F999D607Fa,C40FE90957A7616Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_F4456E408BF15920b;2;2;EC100A4F999D607Fb,C40FE90957A7616Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_F4456E408BF15920c;2;2;EC100A4F999D607Fc,C40FE90957A7616Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_AB3EE52BA5495311a;2;2;F45D493F2C04250Ca,D9E12F0C54F08F09a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_AB3EE52BA5495311b;2;2;F45D493F2C04250Cb,D9E12F0C54F08F09b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_AB3EE52BA5495311c;2;2;F45D493F2C04250Cc,D9E12F0C54F08F09c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_2C5A551A23A0C23Aa;2;2;34CFEF24AACDF931a,5DB35642FC7BC714a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_2C5A551A23A0C23Ab;2;2;34CFEF24AACDF931b,5DB35642FC7BC714b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_2C5A551A23A0C23Ac;2;2;34CFEF24AACDF931c,5DB35642FC7BC714c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_7E04F660771F6D75a;2;2;F36BAE6E5DC3D36Fa,EC00C139B3F76151a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_7E04F660771F6D75b;2;2;F36BAE6E5DC3D36Fb,EC00C139B3F76151b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_7E04F660771F6D75c;2;2;F36BAE6E5DC3D36Fc,EC00C139B3F76151c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_09D76D1BD4C85C44a;2;2;7B5BF00DCB79E36Fa,8AC61C5805667907a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_09D76D1BD4C85C44b;2;2;7B5BF00DCB79E36Fb,8AC61C5805667907b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_09D76D1BD4C85C44c;2;2;7B5BF00DCB79E36Fc,8AC61C5805667907c,
-1,1E+15,0,0,0,,.1,.001,5,

_SW0;_09633065BDE1BE73a;2;2;EB998F2991147947a,97D48338817CE76Ea,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_09633065BDE1BE73b;2;2;EB998F2991147947b,97D48338817CE76Eb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_09633065BDE1BE73c;2;2;EB998F2991147947c,97D48338817CE76Ec,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_56C1CA6D6D56D713a;2;2;863D8457254C3B16a,3232311F943C7523a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_56C1CA6D6D56D713b;2;2;863D8457254C3B16b,3232311F943C7523b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_56C1CA6D6D56D713c;2;2;863D8457254C3B16c,3232311F943C7523c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_C1F8E803DDD4F623a;2;2;02BF5E6C16A10C5Ea,A2B4DA345A658C22a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C1F8E803DDD4F623b;2;2;02BF5E6C16A10C5Eb,A2B4DA345A658C22b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C1F8E803DDD4F623c;2;2;02BF5E6C16A10C5Ec,A2B4DA345A658C22c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_04E2C71BC9304105a;2;2;F30EC4405DCCDC23a,283D3A63E2B34055a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_04E2C71BC9304105b;2;2;F30EC4405DCCDC23b,283D3A63E2B34055b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_04E2C71BC9304105c;2;2;F30EC4405DCCDC23c,283D3A63E2B34055c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_3813D5015EAED739a;2;2;117696009762282Ea,60D79413706CE27Fa,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_3813D5015EAED739b;2;2;117696009762282Eb,60D79413706CE27Fb,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_3813D5015EAED739c;2;2;117696009762282Ec,60D79413706CE27Fc,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_5615106A512CB12Ea;2;2;352EE750CC9F0E12a,DE8475338EC7DD70a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_5615106A512CB12Eb;2;2;352EE750CC9F0E12b,DE8475338EC7DD70b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_5615106A512CB12Ec;2;2;352EE750CC9F0E12c,DE8475338EC7DD70c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_B46E7523434C6A4Da;2;2;58D43143E60E314Ca,A2B4DA345A658C22a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_B46E7523434C6A4Db;2;2;58D43143E60E314Cb,A2B4DA345A658C22b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_B46E7523434C6A4Dc;2;2;58D43143E60E314Cc,A2B4DA345A658C22c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A07DEA100D5C8E1Da;2;2;62BFE060EF00F70Ba,EDD8134417161A29a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A07DEA100D5C8E1Db;2;2;62BFE060EF00F70Bb,EDD8134417161A29b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A07DEA100D5C8E1Dc;2;2;62BFE060EF00F70Bc,EDD8134417161A29c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_DC35413ED87CBC63a;2;2;A917EA31DAB5817Da,CFD9E562BDC2587Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_DC35413ED87CBC63b;2;2;A917EA31DAB5817Db,CFD9E562BDC2587Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_DC35413ED87CBC63c;2;2;A917EA31DAB5817Dc,CFD9E562BDC2587Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_70B6F44E3800FF38a;2;2;A917EA31DAB5817Da,E2DAB632C25AD032a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_70B6F44E3800FF38b;2;2;A917EA31DAB5817Db,E2DAB632C25AD032b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_70B6F44E3800FF38c;2;2;A917EA31DAB5817Dc,E2DAB632C25AD032c,

-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_1F2CFC40997F6559a;2;2;74A80D61D0225C08a,73E6EC01EBD66710a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_1F2CFC40997F6559b;2;2;74A80D61D0225C08b,73E6EC01EBD66710b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_1F2CFC40997F6559c;2;2;74A80D61D0225C08c,73E6EC01EBD66710c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E6519A0DEAA2F442a;2;2;71AF883CCC269279a,373F5D2751838D1Da,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E6519A0DEAA2F442b;2;2;71AF883CCC269279b,373F5D2751838D1Db,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E6519A0DEAA2F442c;2;2;71AF883CCC269279c,373F5D2751838D1Dc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0B302F1BC91DB457a;2;2;CFE55B5DEF93E520a,F70E5A3F25746026a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_0B302F1BC91DB457b;2;2;CFE55B5DEF93E520b,F70E5A3F25746026b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_0B302F1BC91DB457c;2;2;CFE55B5DEF93E520c,F70E5A3F25746026c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_55AA4377EAA5DD46a;2;2;43F8C5234DAC1507a,40C3F902EEB91E56a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_55AA4377EAA5DD46b;2;2;43F8C5234DAC1507b,40C3F902EEB91E56b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_55AA4377EAA5DD46c;2;2;43F8C5234DAC1507c,40C3F902EEB91E56c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9BB4184F8FA8DF42a;2;2;1435785F25869A65a,193F33563A4F710Ea,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9BB4184F8FA8DF42b;2;2;1435785F25869A65b,193F33563A4F710Eb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9BB4184F8FA8DF42c;2;2;1435785F25869A65c,193F33563A4F710Ec,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_369F9211FD987B60a;2;2;F3984E29B32ACF25a,22AA490389CBA843a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_369F9211FD987B60b;2;2;F3984E29B32ACF25b,22AA490389CBA843b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_369F9211FD987B60c;2;2;F3984E29B32ACF25c,22AA490389CBA843c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0AF9040BCD81C74Fa;2;2;B2659007C4867368a,47EB283F13318676a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0AF9040BCD81C74Fb;2;2;B2659007C4867368b,47EB283F13318676b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0AF9040BCD81C74Fc;2;2;B2659007C4867368c,47EB283F13318676c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_28B1832DDC082677a;2;2;C6CECD0F73BF8D63a,BD6EDA06FF1CD976a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_28B1832DDC082677b;2;2;C6CECD0F73BF8D63b,BD6EDA06FF1CD976b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_28B1832DDC082677c;2;2;C6CECD0F73BF8D63c,BD6EDA06FF1CD976c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_2845282AA8E66D04a;2;2;D9E12F0C54F08F09a,CAF2F65604D6D64Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_2845282AA8E66D04b;2;2;D9E12F0C54F08F09b,CAF2F65604D6D64Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_2845282AA8E66D04c;2;2;D9E12F0C54F08F09c,CAF2F65604D6D64Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_2E828728D70D8F32a;2;2;F2B2301D76C1346Ca,ED9C1C4A4EE16C27a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_2E828728D70D8F32b;2;2;F2B2301D76C1346Cb,ED9C1C4A4EE16C27b,
1E+15,1E+15,0,0,0,,.1,.001,5,

_SW0;_2E828728D70D8F32C;2;2;F2B2301D76C1346Cc,ED9C1C4A4EE16C27c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_038DBA3B8BA1274Ba;2;2;A28ED34FBCAE5963a,4BC2A53215E5D162a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_038DBA3B8BA1274Bb;2;2;A28ED34FBCAE5963b,4BC2A53215E5D162b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_038DBA3B8BA1274Bc;2;2;A28ED34FBCAE5963c,4BC2A53215E5D162c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_2A6339205D968C26a;2;2;B323A4660DF0FD52a,F010C746427EAD4Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_2A6339205D968C26b;2;2;B323A4660DF0FD52b,F010C746427EAD4Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_2A6339205D968C26c;2;2;B323A4660DF0FD52c,F010C746427EAD4Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_8B8D5057B88E0D31a;2;2;E3D96616220B5469a,74A80D61D0225C08a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_8B8D5057B88E0D31b;2;2;E3D96616220B5469b,74A80D61D0225C08b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_8B8D5057B88E0D31c;2;2;E3D96616220B5469c,74A80D61D0225C08c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A09AD33F6412753Da;2;2;8242B9323E862241a,2F0A2247C2CBC128a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A09AD33F6412753Db;2;2;8242B9323E862241b,2F0A2247C2CBC128b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A09AD33F6412753Dc;2;2;8242B9323E862241c,2F0A2247C2CBC128c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D7A7B90F59708A18a;2;2;B5CD6D28E8821A22a,1AC5B8204A196F07a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D7A7B90F59708A18b;2;2;B5CD6D28E8821A22b,1AC5B8204A196F07b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D7A7B90F59708A18c;2;2;B5CD6D28E8821A22c,1AC5B8204A196F07c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_731FF62609091138a;2;2;DA4827344EED162Ea,C423EE218A59BC29a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_731FF62609091138b;2;2;DA4827344EED162Eb,C423EE218A59BC29b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_731FF62609091138c;2;2;DA4827344EED162Ec,C423EE218A59BC29c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_850C227D8FCDDC57a;2;2;D3C212544CE75437a,977F7655EC1A022Ea,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_850C227D8FCDDC57b;2;2;D3C212544CE75437b,977F7655EC1A022Eb,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_850C227D8FCDDC57c;2;2;D3C212544CE75437c,977F7655EC1A022Ec,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_FD8239413AA6D21Ba;2;2;588D1D18B6BAE802a,3A830F3E1EF1B848a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_FD8239413AA6D21Bb;2;2;588D1D18B6BAE802b,3A830F3E1EF1B848b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_FD8239413AA6D21Bc;2;2;588D1D18B6BAE802c,3A830F3E1EF1B848c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_10CE691BD64D442Fa;2;2;FDC9D63486D85F78a,373F5D2751838D1Da,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_10CE691BD64D442Fb;2;2;FDC9D63486D85F78b,373F5D2751838D1Db,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_10CE691BD64D442Fc;2;2;FDC9D63486D85F78c,373F5D2751838D1Dc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_2CB4B77312B53205a;2;2;827EFF2D454DFE29a,ADB4F13B1E391241a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_2CB4B77312B53205b;2;2;827EFF2D454DFE29b,ADB4F13B1E391241b,

-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_2CB4B77312B53205c;2;2;827EFF2D454DFE29c,ADB4F13B1E391241c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CDD6B419F055902Aa;2;2;B49BD22CA84C881Ea,6E7617670E246633a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CDD6B419F055902Ab;2;2;B49BD22CA84C881Eb,6E7617670E246633b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CDD6B419F055902Ac;2;2;B49BD22CA84C881Ec,6E7617670E246633c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_EAD7E24E7816D418a;2;2;C450945F45531E29a,F8DBFD62BCDF465Da,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_EAD7E24E7816D418b;2;2;C450945F45531E29b,F8DBFD62BCDF465Db,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_EAD7E24E7816D418c;2;2;C450945F45531E29c,F8DBFD62BCDF465Dc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_EEC7314C9C863978a;2;2;EC00C139B3F76151a,2F6DCC3CEF8BF569a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_EEC7314C9C863978b;2;2;EC00C139B3F76151b,2F6DCC3CEF8BF569b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_EEC7314C9C863978c;2;2;EC00C139B3F76151c,2F6DCC3CEF8BF569c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_95C00E7650262A76a;2;2;68186169E633A724a,B6ABB71E9845EB66a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_95C00E7650262A76b;2;2;68186169E633A724b,B6ABB71E9845EB66b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_95C00E7650262A76c;2;2;68186169E633A724c,B6ABB71E9845EB66c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_61AEFE4EE6480229a;2;2;3483C3759E047072a,CBEA83459B962063a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_61AEFE4EE6480229b;2;2;3483C3759E047072b,CBEA83459B962063b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_61AEFE4EE6480229c;2;2;3483C3759E047072c,CBEA83459B962063c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_1C378343DFC78825a;2;2;01C48F6610722D1Fa,D9E12F0C54F08F09a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_1C378343DFC78825b;2;2;01C48F6610722D1Fb,D9E12F0C54F08F09b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_1C378343DFC78825c;2;2;01C48F6610722D1Fc,D9E12F0C54F08F09c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A54DAC1CA5098B39a;2;2;87507B47331D932Ba,0050334C90DB116Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A54DAC1CA5098B39b;2;2;87507B47331D932Bb,0050334C90DB116Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A54DAC1CA5098B39c;2;2;87507B47331D932Bc,0050334C90DB116Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A3C8896A41E40F23a;2;2;65A10E14B7700B7Fa,D9E12F0C54F08F09a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A3C8896A41E40F23b;2;2;65A10E14B7700B7Fb,D9E12F0C54F08F09b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A3C8896A41E40F23c;2;2;65A10E14B7700B7Fc,D9E12F0C54F08F09c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_BCAD8535DE0DED57a;2;2;0D2B461A6822DE2Da,40294F6B2276B32Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_BCAD8535DE0DED57b;2;2;0D2B461A6822DE2Db,40294F6B2276B32Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_BCAD8535DE0DED57c;2;2;0D2B461A6822DE2Dc,40294F6B2276B32Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_89B2CE3D097D484Fa;2;2;DC2FF6414E0E5D47a,2D81CD178572CE03a,
-1,1E+15,0,0,0,,.1,.001,5,

_SWO;_89B2CE3D097D484Fb;2;2;DC2FF6414E0E5D47b,2D81CD178572CE03b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_89B2CE3D097D484Fc;2;2;DC2FF6414E0E5D47c,2D81CD178572CE03c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_C85726775A63115Ea;2;2;884C6A372F530007a,60D79413706CE27Fa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_C85726775A63115Eb;2;2;884C6A372F530007b,60D79413706CE27Fb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_C85726775A63115Ec;2;2;884C6A372F530007c,60D79413706CE27Fc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_1AE4D65761028D32a;2;2;D3C212544CE75437a,FBD7D244124CFA59a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_1AE4D65761028D32b;2;2;D3C212544CE75437b,FBD7D244124CFA59b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_1AE4D65761028D32c;2;2;D3C212544CE75437c,FBD7D244124CFA59c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_EBA32921FCAEB85Ea;2;2;285FD8270ADF642Aa,1435785F25869A65a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_EBA32921FCAEB85Eb;2;2;285FD8270ADF642Ab,1435785F25869A65b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_EBA32921FCAEB85Ec;2;2;285FD8270ADF642Ac,1435785F25869A65c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_44D5440E238E2633a;2;2;51175F3576CE403Ba,700C7B36D0C2993Da,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_44D5440E238E2633b;2;2;51175F3576CE403Bb,700C7B36D0C2993Db,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_44D5440E238E2633c;2;2;51175F3576CE403Bc,700C7B36D0C2993Dc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_1E40FF7E2608AB67a;2;2;70551608D7F26555a,E81B402AB89F3F61a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_1E40FF7E2608AB67b;2;2;70551608D7F26555b,E81B402AB89F3F61b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_1E40FF7E2608AB67c;2;2;70551608D7F26555c,E81B402AB89F3F61c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_C3E02A43863A7548a;2;2;58D43143E60E314Ca,A6B3CC0D52A13804a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_C3E02A43863A7548b;2;2;58D43143E60E314Cb,A6B3CC0D52A13804b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_C3E02A43863A7548c;2;2;58D43143E60E314Cc,A6B3CC0D52A13804c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_FE68654AE13B8539a;2;2;5DB35642FC7BC714a,53797B5EBC7CB439a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_FE68654AE13B8539b;2;2;5DB35642FC7BC714b,53797B5EBC7CB439b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_FE68654AE13B8539c;2;2;5DB35642FC7BC714c,53797B5EBC7CB439c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_3312D353F540D76Ea;2;2;0E3E3D273D2E0860a,CFD9E562BDC2587Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_3312D353F540D76Eb;2;2;0E3E3D273D2E0860b,CFD9E562BDC2587Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_3312D353F540D76Ec;2;2;0E3E3D273D2E0860c,CFD9E562BDC2587Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4F35056A59B74348a;2;2;D9E12F0C54F08F09a,4AB3A81F1FBFA418a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4F35056A59B74348b;2;2;D9E12F0C54F08F09b,4AB3A81F1FBFA418b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4F35056A59B74348c;2;2;D9E12F0C54F08F09c,4AB3A81F1FBFA418c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_81E2C459A54FDE19a;2;2;6E7617670E246633a,E2B6016E76F4B965a,

-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_81E2C459A54FDE19b;2;2;6E7617670E246633b,E2B6016E76F4B965b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_81E2C459A54FDE19c;2;2;6E7617670E246633c,E2B6016E76F4B965c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9B89FA0B697E426Ba;2;2;47EB283F13318676a,F3984E29B32ACF25a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9B89FA0B697E426Bb;2;2;47EB283F13318676b,F3984E29B32ACF25b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9B89FA0B697E426Bc;2;2;47EB283F13318676c,F3984E29B32ACF25c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_7DCC47386CF2B92Fa;2;2;77C3AB49608B6828a,C2B3562506AFEF45a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_7DCC47386CF2B92Fb;2;2;77C3AB49608B6828b,C2B3562506AFEF45b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_7DCC47386CF2B92Fc;2;2;77C3AB49608B6828c,C2B3562506AFEF45c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_77FD987D4B05C552a;2;2;53D2EA2A06BEB505a,B5FAFF7E2A61B94Fa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_77FD987D4B05C552b;2;2;53D2EA2A06BEB505b,B5FAFF7E2A61B94Fb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_77FD987D4B05C552c;2;2;53D2EA2A06BEB505c,B5FAFF7E2A61B94Fc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_7098BA371AF1FF70a;2;2;F30EC4405DCCDC23a,3258DC79BD22260Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_7098BA371AF1FF70b;2;2;F30EC4405DCCDC23b,3258DC79BD22260Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_7098BA371AF1FF70c;2;2;F30EC4405DCCDC23c,3258DC79BD22260Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9CF09A3EB0C7A04Aa;2;2;ED9C1C4A4EE16C27a,4BC2A53215E5D162a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_9CF09A3EB0C7A04Ab;2;2;ED9C1C4A4EE16C27b,4BC2A53215E5D162b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_9CF09A3EB0C7A04Ac;2;2;ED9C1C4A4EE16C27c,4BC2A53215E5D162c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_288A79549C6C7F5Fa;2;2;AA00BD737AE59A6Ba,BCE6AD5C7ED69C5Ca,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_288A79549C6C7F5Fb;2;2;AA00BD737AE59A6Bb,BCE6AD5C7ED69C5Cb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_288A79549C6C7F5Fc;2;2;AA00BD737AE59A6Bc,BCE6AD5C7ED69C5Cc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_3909BB691B351001a;2;2;5367AB593E89000Da,B323A4660DF0FD52a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_3909BB691B351001b;2;2;5367AB593E89000Db,B323A4660DF0FD52b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_3909BB691B351001c;2;2;5367AB593E89000Dc,B323A4660DF0FD52c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_FFA7661B0D6C9D21a;2;2;C6CECD0F73BF8D63a,24A9D97415E9C166a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_FFA7661B0D6C9D21b;2;2;C6CECD0F73BF8D63b,24A9D97415E9C166b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_FFA7661B0D6C9D21c;2;2;C6CECD0F73BF8D63c,24A9D97415E9C166c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9A23DC2C99D8CA0Fa;2;2;A2B4DA345A658C22a,ABCAF708D79C3737a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9A23DC2C99D8CA0Fb;2;2;A2B4DA345A658C22b,ABCAF708D79C3737b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9A23DC2C99D8CA0Fc;2;2;A2B4DA345A658C22c,ABCAF708D79C3737c,
-1,1E+15,0,0,0,,.1,.001,5,

```

_SWO;_7AD19F42B0AA7B7Ea;2;2;1435785F25869A65a,BB52462116E49328a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_7AD19F42B0AA7B7Eb;2;2;1435785F25869A65b,BB52462116E49328b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_7AD19F42B0AA7B7Ec;2;2;1435785F25869A65c,BB52462116E49328c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_BE13C177A984B057a;2;2;B822583A830FFE22a,6FB3260958FBA52Da,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_BE13C177A984B057b;2;2;B822583A830FFE22b,6FB3260958FBA52Db,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_BE13C177A984B057c;2;2;B822583A830FFE22c,6FB3260958FBA52Dc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_A2D2B0084654550Ea;2;2;0F0CE67017A3A379a,B6518E68578A1D5Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_A2D2B0084654550Eb;2;2;0F0CE67017A3A379b,B6518E68578A1D5Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_A2D2B0084654550Ec;2;2;0F0CE67017A3A379c,B6518E68578A1D5Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_CD9F73226999390Ba;2;2;80251E2EC66C7B63a,884C6A372F530007a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_CD9F73226999390Bb;2;2;80251E2EC66C7B63b,884C6A372F530007b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_CD9F73226999390Bc;2;2;80251E2EC66C7B63c,884C6A372F530007c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_06BB8B7286963E41a;2;2;D9E12F0C54F08F09a,B1F2563B6AC8E20Ea,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_06BB8B7286963E41b;2;2;D9E12F0C54F08F09b,B1F2563B6AC8E20Eb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_06BB8B7286963E41c;2;2;D9E12F0C54F08F09c,B1F2563B6AC8E20Ec,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_37CDA35FD772001Aa;2;2;CB962D2D9F3F543Ca,80251E2EC66C7B63a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_37CDA35FD772001Ab;2;2;CB962D2D9F3F543Cb,80251E2EC66C7B63b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_37CDA35FD772001Ac;2;2;CB962D2D9F3F543Cc,80251E2EC66C7B63c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_6AEDF03852C56914a;2;2;C2B3562506AFEF45a,B15662314D1B733Fa,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_6AEDF03852C56914b;2;2;C2B3562506AFEF45b,B15662314D1B733Fb,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_6AEDF03852C56914c;2;2;C2B3562506AFEF45c,B15662314D1B733Fc,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_6F2E0577BBB93E79a;2;2;E1852D7D480B8044a,C423EE218A59BC29a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_6F2E0577BBB93E79b;2;2;E1852D7D480B8044b,C423EE218A59BC29b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_6F2E0577BBB93E79c;2;2;E1852D7D480B8044c,C423EE218A59BC29c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_A952CF5C4DBCB261a;2;2;ED9C1C4A4EE16C27a,5EF6976722F3E36Ba,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_A952CF5C4DBCB261b;2;2;ED9C1C4A4EE16C27b,5EF6976722F3E36Bb,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_A952CF5C4DBCB261c;2;2;ED9C1C4A4EE16C27c,5EF6976722F3E36Bc,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_3ABC3B55A267C033a;2;2;27591F2A01142231a,FAB63908248C8C38a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_3ABC3B55A267C033b;2;2;27591F2A01142231b,FAB63908248C8C38b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_3ABC3B55A267C033c;2;2;27591F2A01142231c,FAB63908248C8C38c,

```

```

1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_0A7D1B69C5BB9757a;2;2;9794B14836238E03a,6F1DF6336D0AB057a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_0A7D1B69C5BB9757b;2;2;9794B14836238E03b,6F1DF6336D0AB057b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_0A7D1B69C5BB9757c;2;2;9794B14836238E03c,6F1DF6336D0AB057c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_5371C611BA91BB38a;2;2;9CD76D6BE17F0105a,40C3F902EEB91E56a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_5371C611BA91BB38b;2;2;9CD76D6BE17F0105b,40C3F902EEB91E56b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_5371C611BA91BB38c;2;2;9CD76D6BE17F0105c,40C3F902EEB91E56c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4B717514C971BA64a;2;2;F8DBFD62BCDF465Da,7093D25AF7428B35a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4B717514C971BA64b;2;2;F8DBFD62BCDF465Db,7093D25AF7428B35b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4B717514C971BA64c;2;2;F8DBFD62BCDF465Dc,7093D25AF7428B35c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_5288696701816026a;2;2;50AEC4189E40792Aa,1293C157C664DF3Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_5288696701816026b;2;2;50AEC4189E40792Ab,1293C157C664DF3Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_5288696701816026c;2;2;50AEC4189E40792Ac,1293C157C664DF3Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_C164D034E486EF16a;2;2;62BFE060EF00F70Ba,A83A4716E4A1F933a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_C164D034E486EF16b;2;2;62BFE060EF00F70Bb,A83A4716E4A1F933b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_C164D034E486EF16c;2;2;62BFE060EF00F70Bc,A83A4716E4A1F933c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_07D4081074EEE351a;2;2;51175F3576CE403Ba,8AF1F339B6477A1Ca,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_07D4081074EEE351b;2;2;51175F3576CE403Bb,8AF1F339B6477A1Cb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_07D4081074EEE351c;2;2;51175F3576CE403Bc,8AF1F339B6477A1Cc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_2293E365160F3B26a;2;2;ECC9626CDC4CA76Ca,C5EC267423E39F1Fa,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_2293E365160F3B26b;2;2;ECC9626CDC4CA76Cb,C5EC267423E39F1Fb,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_2293E365160F3B26c;2;2;ECC9626CDC4CA76Cc,C5EC267423E39F1Fc,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_59267122B32D645Fa;2;2;D2D640537EB38F0Aa,CF721A2D7629203Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_59267122B32D645Fb;2;2;D2D640537EB38F0Ab,CF721A2D7629203Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_59267122B32D645Fc;2;2;D2D640537EB38F0Ac,CF721A2D7629203Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_CFC1D331A447DE21a;2;2;CA7B405350470373a,03D9E90087EB3120a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_CFC1D331A447DE21b;2;2;CA7B405350470373b,03D9E90087EB3120b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_CFC1D331A447DE21c;2;2;CA7B405350470373c,03D9E90087EB3120c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_034AE26D62A4C75Ba;2;2;40C3F902EEB91E56a,2FF83207E206145Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_034AE26D62A4C75Bb;2;2;40C3F902EEB91E56b,2FF83207E206145Bb,
-1,1E+15,0,0,0,,.1,.001,5,

```

_SW0;_034AE26D62A4C75Bc;2;2;40C3F902EEB91E56c,2FF83207E206145Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CF27CD5F72BC0A5Aa;2;2;C423EE218A59BC29a,E0947E2A0145C963a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CF27CD5F72BC0A5Ab;2;2;C423EE218A59BC29b,E0947E2A0145C963b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CF27CD5F72BC0A5Ac;2;2;C423EE218A59BC29c,E0947E2A0145C963c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_F5F9EB44BFD33542a;2;2;13C79967ABDE7C57a,C6CECD0F73BF8D63a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_F5F9EB44BFD33542b;2;2;13C79967ABDE7C57b,C6CECD0F73BF8D63b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_F5F9EB44BFD33542c;2;2;13C79967ABDE7C57c,C6CECD0F73BF8D63c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A495734608C5C248a;2;2;977F7655EC1A022Ea,FBD7D244124CFA59a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A495734608C5C248b;2;2;977F7655EC1A022Eb,FBD7D244124CFA59b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A495734608C5C248c;2;2;977F7655EC1A022Ec,FBD7D244124CFA59c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_3F51DF5B40914749a;2;2;9277683CC735A247a,A5FC1D6E9BA8E600a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_3F51DF5B40914749b;2;2;9277683CC735A247b,A5FC1D6E9BA8E600b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_3F51DF5B40914749c;2;2;9277683CC735A247c,A5FC1D6E9BA8E600c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D809AD5C149ADD55a;2;2;40C3F902EEB91E56a,6F56FC25C94B7A14a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D809AD5C149ADD55b;2;2;40C3F902EEB91E56b,6F56FC25C94B7A14b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D809AD5C149ADD55c;2;2;40C3F902EEB91E56c,6F56FC25C94B7A14c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E50148331B8B2E0Aa;2;2;C6CECD0F73BF8D63a,9BF8F2615236895Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E50148331B8B2E0Ab;2;2;C6CECD0F73BF8D63b,9BF8F2615236895Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E50148331B8B2E0Ac;2;2;C6CECD0F73BF8D63c,9BF8F2615236895Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_119AE700AE9ED422a;2;2;940789700C54B077a,BD58C9514D5B611Ea,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_119AE700AE9ED422b;2;2;940789700C54B077b,BD58C9514D5B611Eb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_119AE700AE9ED422c;2;2;940789700C54B077c,BD58C9514D5B611Ec,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_DDAFCA4A28BDA944a;2;2;B54FBF48D898A61Aa,97745117104DB150a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_DDAFCA4A28BDA944b;2;2;B54FBF48D898A61Ab,97745117104DB150b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_DDAFCA4A28BDA944c;2;2;B54FBF48D898A61Ac,97745117104DB150c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D65DAB0F277BA379a;2;2;2A821144EDE5A226a,D9E12F0C54F08F09a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D65DAB0F277BA379b;2;2;2A821144EDE5A226b,D9E12F0C54F08F09b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D65DAB0F277BA379c;2;2;2A821144EDE5A226c,D9E12F0C54F08F09c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_BFD04A12AE81E50Aa;2;2;A83A4716E4A1F933a,C50F745C1E57492Ca,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_BFD04A12AE81E50Ab;2;2;A83A4716E4A1F933b,C50F745C1E57492Cb,

-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_BFD04A12AE81E50Ac;2;2;A83A4716E4A1F933c,C50F745C1E57492Cc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_10D43F334007790Ba;2;2;CF6C3F4DF2AD1749a,09DFA371E9B9821Ea,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_10D43F334007790Bb;2;2;CF6C3F4DF2AD1749b,09DFA371E9B9821Eb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_10D43F334007790Bc;2;2;CF6C3F4DF2AD1749c,09DFA371E9B9821Ec,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_7A579039238DBB2Da;2;2;A28ED34FBCAE5963a,5EF6976722F3E36Ba,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_7A579039238DBB2Db;2;2;A28ED34FBCAE5963b,5EF6976722F3E36Bb,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_7A579039238DBB2Dc;2;2;A28ED34FBCAE5963c,5EF6976722F3E36Bc,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_72209B7BE57B941Ea;2;2;66D858026E93426Ba,5CDC140FD318135Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_72209B7BE57B941Eb;2;2;66D858026E93426Bb,5CDC140FD318135Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_72209B7BE57B941Ec;2;2;66D858026E93426Bc,5CDC140FD318135Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_E991DD4601F00D27a;2;2;7897704E05770161a,B6518E68578A1D5Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_E991DD4601F00D27b;2;2;7897704E05770161b,B6518E68578A1D5Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_E991DD4601F00D27c;2;2;7897704E05770161c,B6518E68578A1D5Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_25F2556C85E14F69a;2;2;371E5474FDFA7C2Aa,13E29A4376B95F28a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_25F2556C85E14F69b;2;2;371E5474FDFA7C2Ab,13E29A4376B95F28b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_25F2556C85E14F69c;2;2;371E5474FDFA7C2Ac,13E29A4376B95F28c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4775FE126F0E436Aa;2;2;B6ABB71E9845EB66a,DB4DD01F7A57C570a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4775FE126F0E436Ab;2;2;B6ABB71E9845EB66b,DB4DD01F7A57C570b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4775FE126F0E436Ac;2;2;B6ABB71E9845EB66c,DB4DD01F7A57C570c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_36D5015FB4479209a;2;2;2C1383656D1E417Aa,0BD8B02E0C91C62Fa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_36D5015FB4479209b;2;2;2C1383656D1E417Ab,0BD8B02E0C91C62Fb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_36D5015FB4479209c;2;2;2C1383656D1E417Ac,0BD8B02E0C91C62Fc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_975D997C3CDD4635a;2;2;59E4567D6A66CE6Da,C2B3562506AFEF45a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_975D997C3CDD4635b;2;2;59E4567D6A66CE6Db,C2B3562506AFEF45b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_975D997C3CDD4635c;2;2;59E4567D6A66CE6Dc,C2B3562506AFEF45c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_AC26E92171DACE4Da;2;2;489F1F498213AD67a,BD2D2A07F2FA2923a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_AC26E92171DACE4Db;2;2;489F1F498213AD67b,BD2D2A07F2FA2923b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_AC26E92171DACE4Dc;2;2;489F1F498213AD67c,BD2D2A07F2FA2923c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_31E5B90D2928B55Ba;2;2;58D43143E60E314Ca,8AF1F339B6477A1Ca,
-1,1E+15,0,0,0,,.1,.001,5,

_SW0;_31E5B90D2928B55Bb;2;2;58D43143E60E314Cb,8AF1F339B6477A1Cb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_31E5B90D2928B55Bc;2;2;58D43143E60E314Cc,8AF1F339B6477A1Cc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9490DB3ABB2EA574a;2;2;B49BD22CA84C881Ea,13E29A4376B95F28a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9490DB3ABB2EA574b;2;2;B49BD22CA84C881Eb,13E29A4376B95F28b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9490DB3ABB2EA574c;2;2;B49BD22CA84C881Ec,13E29A4376B95F28c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_B869FC6B7164DA17a;2;2;D9BCE86619781D1Aa,0050334C90DB116Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_B869FC6B7164DA17b;2;2;D9BCE86619781D1Ab,0050334C90DB116Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_B869FC6B7164DA17c;2;2;D9BCE86619781D1Ac,0050334C90DB116Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_7FFE58692DDF634Ba;2;2;183B913915598B02a,FE8B2715FEE0F112a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_7FFE58692DDF634Bb;2;2;183B913915598B02b,FE8B2715FEE0F112b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_7FFE58692DDF634Bc;2;2;183B913915598B02c,FE8B2715FEE0F112c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_7DF1A07BF19D5A0Ea;2;2;C192583DEFC58611a,66D858026E93426Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_7DF1A07BF19D5A0Eb;2;2;C192583DEFC58611b,66D858026E93426Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_7DF1A07BF19D5A0Ec;2;2;C192583DEFC58611c,66D858026E93426Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_1EA21F5D85FB5054a;2;2;5040590A34818C37a,CF721A2D7629203Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_1EA21F5D85FB5054b;2;2;5040590A34818C37b,CF721A2D7629203Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_1EA21F5D85FB5054c;2;2;5040590A34818C37c,CF721A2D7629203Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0BAC685DB1D5126Ba;2;2;80251E2EC66C7B63a,60D79413706CE27Fa,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0BAC685DB1D5126Bb;2;2;80251E2EC66C7B63b,60D79413706CE27Fb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0BAC685DB1D5126Bc;2;2;80251E2EC66C7B63c,60D79413706CE27Fc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_8C3C6416E42BAD31a;2;2;A80A65213C7B5751a,66D858026E93426Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_8C3C6416E42BAD31b;2;2;A80A65213C7B5751b,66D858026E93426Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_8C3C6416E42BAD31c;2;2;A80A65213C7B5751c,66D858026E93426Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_09C44215B72E4614a;2;2;A28ED34FBCAE5963a,ED9C1C4A4EE16C27a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_09C44215B72E4614b;2;2;A28ED34FBCAE5963b,ED9C1C4A4EE16C27b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_09C44215B72E4614c;2;2;A28ED34FBCAE5963c,ED9C1C4A4EE16C27c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_0EF3CD39C134EE44a;2;2;C203F07798607476a,7723D33730FEC57Ba,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_0EF3CD39C134EE44b;2;2;C203F07798607476b,7723D33730FEC57Bb,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_0EF3CD39C134EE44c;2;2;C203F07798607476c,7723D33730FEC57Bc,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_485C003CBF1D316Aa;2;2;B2D87B3EFC54347Da,557BF85536892712a,

-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_485C003CBF1D316Ab;2;2;B2D87B3EFC54347Db,557BF85536892712b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_485C003CBF1D316Ac;2;2;B2D87B3EFC54347Dc,557BF85536892712c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_49410737CB2BC321a;2;2;A14C0B3EA4F20A1Aa,9794B14836238E03a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_49410737CB2BC321b;2;2;A14C0B3EA4F20A1Ab,9794B14836238E03b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_49410737CB2BC321c;2;2;A14C0B3EA4F20A1Ac,9794B14836238E03c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_869486603ABC0456a;2;2;6BCED8475F1D0371a,59E4567D6A66CE6Da,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_869486603ABC0456b;2;2;6BCED8475F1D0371b,59E4567D6A66CE6Db,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_869486603ABC0456c;2;2;6BCED8475F1D0371c,59E4567D6A66CE6Dc,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_E0BB4E1B3A602006a;2;2;0983D076B975E85Da,6050856F99EC9D02a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_E0BB4E1B3A602006b;2;2;0983D076B975E85Db,6050856F99EC9D02b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_E0BB4E1B3A602006c;2;2;0983D076B975E85Dc,6050856F99EC9D02c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4C22A37C5ED4C04Aa;2;2;F2B2301D76C1346Ca,A28ED34FBCAE5963a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_4C22A37C5ED4C04Ab;2;2;F2B2301D76C1346Cb,A28ED34FBCAE5963b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_4C22A37C5ED4C04Ac;2;2;F2B2301D76C1346Cc,A28ED34FBCAE5963c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_428880050A3EFE4Fa;2;2;5DB35642FC7BC714a,FDA7FC5780D5427Ea,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_428880050A3EFE4Fb;2;2;5DB35642FC7BC714b,FDA7FC5780D5427Eb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_428880050A3EFE4Fc;2;2;5DB35642FC7BC714c,FDA7FC5780D5427Ec,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_94B9AB0AF0DD7721a;2;2;8242B9323E862241a,B560A675909BB879a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_94B9AB0AF0DD7721b;2;2;8242B9323E862241b,B560A675909BB879b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_94B9AB0AF0DD7721c;2;2;8242B9323E862241c,B560A675909BB879c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_B2462F608CB90004a;2;2;13E29A4376B95F28a,4BFA5623F5D89002a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_B2462F608CB90004b;2;2;13E29A4376B95F28b,4BFA5623F5D89002b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_B2462F608CB90004c;2;2;13E29A4376B95F28c,4BFA5623F5D89002c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_84ECAD320920F379a;2;2;C423EE218A59BC29a,75D66212A0924429a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_84ECAD320920F379b;2;2;C423EE218A59BC29b,75D66212A0924429b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_84ECAD320920F379c;2;2;C423EE218A59BC29c,75D66212A0924429c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4AA7B24CC6C22F4Ca;2;2;2D81CD178572CE03a,A4DD1617B00EFB3Aa,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_4AA7B24CC6C22F4Cb;2;2;2D81CD178572CE03b,A4DD1617B00EFB3Ab,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_4AA7B24CC6C22F4Cc;2;2;2D81CD178572CE03c,A4DD1617B00EFB3Ac,
1E+15,1E+15,0,0,0,,.1,.001,5,

_SWO;_BD322A081910FB67a;2;2;F9DFFA5932869516a,3483C3759E047072a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_BD322A081910FB67b;2;2;F9DFFA5932869516b,3483C3759E047072b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_BD322A081910FB67c;2;2;F9DFFA5932869516c,3483C3759E047072c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_45D06A4C3A313C4Ca;2;2;46134844A6699753a,0BD8B02E0C91C62Fa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_45D06A4C3A313C4Cb;2;2;46134844A6699753b,0BD8B02E0C91C62Fb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_45D06A4C3A313C4Cc;2;2;46134844A6699753c,0BD8B02E0C91C62Fc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_6ABD5959B8CE4966a;2;2;959FAE2E97227A68a,588D1D18B6BAE802a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_6ABD5959B8CE4966b;2;2;959FAE2E97227A68b,588D1D18B6BAE802b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_6ABD5959B8CE4966c;2;2;959FAE2E97227A68c,588D1D18B6BAE802c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_2830F30479233926a;2;2;94A6E95316AF6535a,C40FE90957A7616Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_2830F30479233926b;2;2;94A6E95316AF6535b,C40FE90957A7616Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_2830F30479233926c;2;2;94A6E95316AF6535c,C40FE90957A7616Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_E3B9F23D050D3455a;2;2;CCC832566093F215a,BD58C9514D5B611Ea,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_E3B9F23D050D3455b;2;2;CCC832566093F215b,BD58C9514D5B611Eb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_E3B9F23D050D3455c;2;2;CCC832566093F215c,BD58C9514D5B611Ec,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_7E130B4CECA70241a;2;2;CF721A2D7629203Aa,6A37E80B57F1F456a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_7E130B4CECA70241b;2;2;CF721A2D7629203Ab,6A37E80B57F1F456b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_7E130B4CECA70241c;2;2;CF721A2D7629203Ac,6A37E80B57F1F456c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_46130031F4AD8050a;2;2;CF721A2D7629203Aa,E5934D31D9EDCB62a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_46130031F4AD8050b;2;2;CF721A2D7629203Ab,E5934D31D9EDCB62b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_46130031F4AD8050c;2;2;CF721A2D7629203Ac,E5934D31D9EDCB62c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_D334982FED494E41a;2;2;AC0D116F0F141E3Aa,CF721A2D7629203Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_D334982FED494E41b;2;2;AC0D116F0F141E3Ab,CF721A2D7629203Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_D334982FED494E41c;2;2;AC0D116F0F141E3Ac,CF721A2D7629203Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4EBF504DCA893814a;2;2;371E5474FDFA7C2Aa,71AF883CCC269279a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4EBF504DCA893814b;2;2;371E5474FDFA7C2Ab,71AF883CCC269279b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_4EBF504DCA893814c;2;2;371E5474FDFA7C2Ac,71AF883CCC269279c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_30AC18650CDC7B60a;2;2;7148F27A5B75E40Ea,616275244D7CAE77a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_30AC18650CDC7B60b;2;2;7148F27A5B75E40Eb,616275244D7CAE77b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_30AC18650CDC7B60c;2;2;7148F27A5B75E40Ec,616275244D7CAE77c,

-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_BA1AB253A0851325a;2;2;98AF910BECCC8E45a,0818100980654841a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_BA1AB253A0851325b;2;2;98AF910BECCC8E45b,0818100980654841b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_BA1AB253A0851325c;2;2;98AF910BECCC8E45c,0818100980654841c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_79CD095190C97A1Fa;2;2;BD58C9514D5B611Ea,C005890625AA7A17a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_79CD095190C97A1Fb;2;2;BD58C9514D5B611Eb,C005890625AA7A17b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_79CD095190C97A1Fc;2;2;BD58C9514D5B611Ec,C005890625AA7A17c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E6D22A23D513D339a;2;2;40294F6B2276B32Ba,B878E07C3567CA74a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E6D22A23D513D339b;2;2;40294F6B2276B32Bb,B878E07C3567CA74b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E6D22A23D513D339c;2;2;40294F6B2276B32Bc,B878E07C3567CA74c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_82FCC91DAC8EA370a;2;2;40294F6B2276B32Ba,37A392019AEC8577a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_82FCC91DAC8EA370b;2;2;40294F6B2276B32Bb,37A392019AEC8577b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_82FCC91DAC8EA370c;2;2;40294F6B2276B32Bc,37A392019AEC8577c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9726F96CA0FDD055a;2;2;7B5BF00DCB79E36Fa,22618876924EB524a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9726F96CA0FDD055b;2;2;7B5BF00DCB79E36Fb,22618876924EB524b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9726F96CA0FDD055c;2;2;7B5BF00DCB79E36Fc,22618876924EB524c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_72DF2163EAAD013Fa;2;2;66D858026E93426Ba,0B4AA268392D925Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_72DF2163EAAD013Fb;2;2;66D858026E93426Bb,0B4AA268392D925Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_72DF2163EAAD013Fc;2;2;66D858026E93426Bc,0B4AA268392D925Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_7343F35776AA0D44a;2;2;A5FC1D6E9BA8E600a,A833467508CFD705a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_7343F35776AA0D44b;2;2;A5FC1D6E9BA8E600b,A833467508CFD705b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_7343F35776AA0D44c;2;2;A5FC1D6E9BA8E600c,A833467508CFD705c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_5AB80E1DA3A49E26a;2;2;7346E11A6B5C9309a,A28ED34FBCAE5963a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_5AB80E1DA3A49E26b;2;2;7346E11A6B5C9309b,A28ED34FBCAE5963b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_5AB80E1DA3A49E26c;2;2;7346E11A6B5C9309c,A28ED34FBCAE5963c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_F6526C58EA1FEB34a;2;2;CFE55B5DEF93E520a,F70E5A3F25746026a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_F6526C58EA1FEB34b;2;2;CFE55B5DEF93E520b,F70E5A3F25746026b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_F6526C58EA1FEB34c;2;2;CFE55B5DEF93E520c,F70E5A3F25746026c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_93A6864C358F895Fa;2;2;557BF85536892712a,EC5A647CF0837657a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_93A6864C358F895Fb;2;2;557BF85536892712b,EC5A647CF0837657b,
-1,1E+15,0,0,0,,.1,.001,5,

```

_SWO;_93A6864C358F895Fc;2;2;557BF85536892712c,EC5A647CF0837657c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9243E6247159D800a;2;2;DE8475338EC7DD70a,96F75313285FFE2Ca,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9243E6247159D800b;2;2;DE8475338EC7DD70b,96F75313285FFE2Cb,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9243E6247159D800c;2;2;DE8475338EC7DD70c,96F75313285FFE2Cc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_AA4BD27C87CEFD28a;2;2;7723D33730FEC57Ba,CA7B405350470373a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_AA4BD27C87CEFD28b;2;2;7723D33730FEC57Bb,CA7B405350470373b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_AA4BD27C87CEFD28c;2;2;7723D33730FEC57Bc,CA7B405350470373c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_FA0C384BFEB4282Fa;2;2;9277683CC735A247a,5A66AD6903F3667Da,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_FA0C384BFEB4282Fb;2;2;9277683CC735A247b,5A66AD6903F3667Db,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_FA0C384BFEB4282Fc;2;2;9277683CC735A247c,5A66AD6903F3667Dc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9A9FD63CCD01310Ea;2;2;6FB3260958FBA52Da,72D0BF5B263F8F73a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9A9FD63CCD01310Eb;2;2;6FB3260958FBA52Db,72D0BF5B263F8F73b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9A9FD63CCD01310Ec;2;2;6FB3260958FBA52Dc,72D0BF5B263F8F73c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_51AE654BDB07FD21a;2;2;C6C1406A3194A954a,489F1F498213AD67a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_51AE654BDB07FD21b;2;2;C6C1406A3194A954b,489F1F498213AD67b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_51AE654BDB07FD21c;2;2;C6C1406A3194A954c,489F1F498213AD67c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_B7CD292545641066a;2;2;A26EC97EE927216Fa,51D8F76FC7FF4025a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_B7CD292545641066b;2;2;A26EC97EE927216Fb,51D8F76FC7FF4025b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_B7CD292545641066c;2;2;A26EC97EE927216Fc,51D8F76FC7FF4025c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_22C0D76F887E4838a;2;2;F30EC4405DCCDC23a,F3984E29B32ACF25a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_22C0D76F887E4838b;2;2;F30EC4405DCCDC23b,F3984E29B32ACF25b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_22C0D76F887E4838c;2;2;F30EC4405DCCDC23c,F3984E29B32ACF25c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9916553ABEB6D20Ca;2;2;AAAA19740986F679a,709D5530F4FC4B0Da,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9916553ABEB6D20Cb;2;2;AAAA19740986F679b,709D5530F4FC4B0Db,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_9916553ABEB6D20Cc;2;2;AAAA19740986F679c,709D5530F4FC4B0Dc,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_44E3D45AE9B0BB0Aa;2;2;66D858026E93426Ba,58F6211107750854a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_44E3D45AE9B0BB0Ab;2;2;66D858026E93426Bb,58F6211107750854b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_44E3D45AE9B0BB0Ac;2;2;66D858026E93426Bc,58F6211107750854c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_FF681159FE08BC3Ba;2;2;3258DC79BD22260Aa,47EB283F13318676a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_FF681159FE08BC3Bb;2;2;3258DC79BD22260Ab,47EB283F13318676b,

```

-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_FF681159FE08BC3Bc;2;2;3258DC79BD22260Ac,47EB283F13318676c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_66991A79BC69A674a;2;2;C40FE90957A7616Aa,629E822A7403A252a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_66991A79BC69A674b;2;2;C40FE90957A7616Ab,629E822A7403A252b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_66991A79BC69A674c;2;2;C40FE90957A7616Ac,629E822A7403A252c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_6D23F856603FE16Da;2;2;581948073631FB1Da,50AEC4189E40792Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_6D23F856603FE16Db;2;2;581948073631FB1Db,50AEC4189E40792Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_6D23F856603FE16Dc;2;2;581948073631FB1Dc,50AEC4189E40792Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_25E15A3B2C343D73a;2;2;66D858026E93426Ba,828D8E0B9ADA0A2Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_25E15A3B2C343D73b;2;2;66D858026E93426Bb,828D8E0B9ADA0A2Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_25E15A3B2C343D73c;2;2;66D858026E93426Bc,828D8E0B9ADA0A2Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_85811C73EE864E0Da;2;2;AA37B14CAF65CA1Ba,10D90C4694A96418a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_85811C73EE864E0Db;2;2;AA37B14CAF65CA1Bb,10D90C4694A96418b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_85811C73EE864E0Dc;2;2;AA37B14CAF65CA1Bc,10D90C4694A96418c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SWO;_8DA1EC6DF294CF19a;2;2;FAB63908248C8C38a,27591F2A01142231a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_8DA1EC6DF294CF19b;2;2;FAB63908248C8C38b,27591F2A01142231b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_8DA1EC6DF294CF19c;2;2;FAB63908248C8C38c,27591F2A01142231c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_BCDCC7182E62E029a;2;2;C6E19262B8856D01a,FE8B2715FEE0F112a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_BCDCC7182E62E029b;2;2;C6E19262B8856D01b,FE8B2715FEE0F112b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_BCDCC7182E62E029c;2;2;C6E19262B8856D01c,FE8B2715FEE0F112c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_3815CF79EE739235a;2;2;4C87317B0036E835a,B6518E68578A1D5Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_3815CF79EE739235b;2;2;4C87317B0036E835b,B6518E68578A1D5Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_3815CF79EE739235c;2;2;4C87317B0036E835c,B6518E68578A1D5Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_A2302855FE694B02a;2;2;66D858026E93426Ba,18A575649CB8C025a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_A2302855FE694B02b;2;2;66D858026E93426Bb,18A575649CB8C025b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_A2302855FE694B02c;2;2;66D858026E93426Bc,18A575649CB8C025c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_D00D08242D9C3A32a;2;2;709D5530F4FC4B0Da,74A80D61D0225C08a,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_D00D08242D9C3A32b;2;2;709D5530F4FC4B0Db,74A80D61D0225C08b,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_D00D08242D9C3A32c;2;2;709D5530F4FC4B0Dc,74A80D61D0225C08c,
-1,1E+15,0,0,0,,.1,.001,5,
_SWO;_02A41C798138F019a;2;2;B822583A830FFE22a,02BF5E6C16A10C5Ea,
-1,1E+15,0,0,0,,.1,.001,5,

_SW0;_02A41C798138F019b;2;2;B822583A830FFE22b,02BF5E6C16A10C5Eb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_02A41C798138F019c;2;2;B822583A830FFE22c,02BF5E6C16A10C5Ec,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_EFF4A0778B170656a;2;2;B6518E68578A1D5Aa,FEAD364EBBABC91Ca,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_EFF4A0778B170656b;2;2;B6518E68578A1D5Ab,FEAD364EBBABC91Cb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_EFF4A0778B170656c;2;2;B6518E68578A1D5Ac,FEAD364EBBABC91Cc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_920EFA5BDC74D14Aa;2;2;5DB35642FC7BC714a,2F8C384C704A8879a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_920EFA5BDC74D14Ab;2;2;5DB35642FC7BC714b,2F8C384C704A8879b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_920EFA5BDC74D14Ac;2;2;5DB35642FC7BC714c,2F8C384C704A8879c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_438C4C04CFF38C09a;2;2;9E0ECD16EC5EE41Ba,73E6EC01EBD66710a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_438C4C04CFF38C09b;2;2;9E0ECD16EC5EE41Bb,73E6EC01EBD66710b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_438C4C04CFF38C09c;2;2;9E0ECD16EC5EE41Bc,73E6EC01EBD66710c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0FA4ED4878594E24a;2;2;C423EE218A59BC29a,6E323109C3F7FA54a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0FA4ED4878594E24b;2;2;C423EE218A59BC29b,6E323109C3F7FA54b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0FA4ED4878594E24c;2;2;C423EE218A59BC29c,6E323109C3F7FA54c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0395D8211132E137a;2;2;0818100980654841a,98AF910BECCC8E45a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0395D8211132E137b;2;2;0818100980654841b,98AF910BECCC8E45b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0395D8211132E137c;2;2;0818100980654841c,98AF910BECCC8E45c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_DDC69E14A0AF5262a;2;2;0BD8B02E0C91C62Fa,8B54113CB432E22Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_DDC69E14A0AF5262b;2;2;0BD8B02E0C91C62Fb,8B54113CB432E22Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_DDC69E14A0AF5262c;2;2;0BD8B02E0C91C62Fc,8B54113CB432E22Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_F427B950F265106Ba;2;2;C6CECD0F73BF8D63a,6A477724EAE7777Ca,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_F427B950F265106Bb;2;2;C6CECD0F73BF8D63b,6A477724EAE7777Cb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_F427B950F265106Bc;2;2;C6CECD0F73BF8D63c,6A477724EAE7777Cc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_544222473C39616Da;2;2;A83A4716E4A1F933a,C6E19262B8856D01a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_544222473C39616Db;2;2;A83A4716E4A1F933b,C6E19262B8856D01b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_544222473C39616Dc;2;2;A83A4716E4A1F933c,C6E19262B8856D01c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C2550924D9DC3977a;2;2;C1C7F47F1682D105a,2D81CD178572CE03a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C2550924D9DC3977b;2;2;C1C7F47F1682D105b,2D81CD178572CE03b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C2550924D9DC3977c;2;2;C1C7F47F1682D105c,2D81CD178572CE03c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9EA87052B1D10412a;2;2;F2750720CF356304a,C005890625AA7A17a,

-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9EA87052B1D10412b;2;2;F2750720CF356304b,C005890625AA7A17b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9EA87052B1D10412c;2;2;F2750720CF356304c,C005890625AA7A17c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9BA1194AE4E46E16a;2;2;1B3F8E57788BF609a,66D858026E93426Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9BA1194AE4E46E16b;2;2;1B3F8E57788BF609b,66D858026E93426Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9BA1194AE4E46E16c;2;2;1B3F8E57788BF609c,66D858026E93426Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_44C72959AD898603a;2;2;1EA0A91354C96133a,8242B9323E862241a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_44C72959AD898603b;2;2;1EA0A91354C96133b,8242B9323E862241b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_44C72959AD898603c;2;2;1EA0A91354C96133c,8242B9323E862241c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CE2EA170CC7EE00Ca;2;2;2F56BA6B50A4D74Ba,09DFA371E9B9821Ea,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CE2EA170CC7EE00Cb;2;2;2F56BA6B50A4D74Bb,09DFA371E9B9821Eb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CE2EA170CC7EE00Cc;2;2;2F56BA6B50A4D74Bc,09DFA371E9B9821Ec,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_859BB02C4E3E0F30a;2;2;7346E11A6B5C9309a,ED9C1C4A4EE16C27a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_859BB02C4E3E0F30b;2;2;7346E11A6B5C9309b,ED9C1C4A4EE16C27b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_859BB02C4E3E0F30c;2;2;7346E11A6B5C9309c,ED9C1C4A4EE16C27c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_E78A971B42DAF050a;2;2;810D3B4E4C730B2Da,8242B9323E862241a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E78A971B42DAF050b;2;2;810D3B4E4C730B2Db,8242B9323E862241b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E78A971B42DAF050c;2;2;810D3B4E4C730B2Dc,8242B9323E862241c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_F3278545FB43750Ea;2;2;4FFF1B064FF0B60Aa,0050334C90DB116Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_F3278545FB43750Eb;2;2;4FFF1B064FF0B60Ab,0050334C90DB116Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_F3278545FB43750Ec;2;2;4FFF1B064FF0B60Ac,0050334C90DB116Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_544AEE689CFD9010a;2;2;C2B3562506AFEF45a,D77D5958909FAB6Da,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_544AEE689CFD9010b;2;2;C2B3562506AFEF45b,D77D5958909FAB6Db,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_544AEE689CFD9010c;2;2;C2B3562506AFEF45c,D77D5958909FAB6Dc,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_261DC676A3E71558a;2;2;BCE6AD5C7ED69C5Ca,DE8475338EC7DD70a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_261DC676A3E71558b;2;2;BCE6AD5C7ED69C5Cb,DE8475338EC7DD70b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_261DC676A3E71558c;2;2;BCE6AD5C7ED69C5Cc,DE8475338EC7DD70c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_434C4327A29E3967a;2;2;93406A72C225963Fa,1E10C745D6204208a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_434C4327A29E3967b;2;2;93406A72C225963Fb,1E10C745D6204208b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_434C4327A29E3967c;2;2;93406A72C225963Fc,1E10C745D6204208c,
-1,1E+15,0,0,0,,.1,.001,5,

_SW0;_B938FB3809297726a;2;2;ED9C1C4A4EE16C27a,7FB36165489E8C75a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_B938FB3809297726b;2;2;ED9C1C4A4EE16C27b,7FB36165489E8C75b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_B938FB3809297726c;2;2;ED9C1C4A4EE16C27c,7FB36165489E8C75c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_A3ABDD69E1583A5Ca;2;2;1435785F25869A65a,DEE8F26AFE0A4021a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A3ABDD69E1583A5Cb;2;2;1435785F25869A65b,DEE8F26AFE0A4021b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A3ABDD69E1583A5Cc;2;2;1435785F25869A65c,DEE8F26AFE0A4021c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C5F27D6E74FD8247a;2;2;3483C3759E047072a,81A2B223A8DB4538a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C5F27D6E74FD8247b;2;2;3483C3759E047072b,81A2B223A8DB4538b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C5F27D6E74FD8247c;2;2;3483C3759E047072c,81A2B223A8DB4538c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_71F51F3AAF1A6A5Fa;2;2;3C4CB519C7D48E56a,557BF85536892712a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_71F51F3AAF1A6A5Fb;2;2;3C4CB519C7D48E56b,557BF85536892712b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_71F51F3AAF1A6A5Fc;2;2;3C4CB519C7D48E56c,557BF85536892712c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_54531D1D60E84212a;2;2;A4DD1617B00EFB3Aa,B7389473D674782Ca,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_54531D1D60E84212b;2;2;A4DD1617B00EFB3Ab,B7389473D674782Cb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_54531D1D60E84212c;2;2;A4DD1617B00EFB3Ac,B7389473D674782Cc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_84E1692756CFF468a;2;2;6BCED8475F1D0371a,B15662314D1B733Fa,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_84E1692756CFF468b;2;2;6BCED8475F1D0371b,B15662314D1B733Fb,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_84E1692756CFF468c;2;2;6BCED8475F1D0371c,B15662314D1B733Fc,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_81E8F8719632BE7Aa;2;2;EC00C139B3F76151a,50AEC4189E40792Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_81E8F8719632BE7Ab;2;2;EC00C139B3F76151b,50AEC4189E40792Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_81E8F8719632BE7Ac;2;2;EC00C139B3F76151c,50AEC4189E40792Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_12628B43A2DB5903a;2;2;93406A72C225963Fa,62BFE060EF00F70Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_12628B43A2DB5903b;2;2;93406A72C225963Fb,62BFE060EF00F70Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_12628B43A2DB5903c;2;2;93406A72C225963Fc,62BFE060EF00F70Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_B6CC732FCAB91D57a;2;2;A833467508CFD705a,9277683CC735A247a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_B6CC732FCAB91D57b;2;2;A833467508CFD705b,9277683CC735A247b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_B6CC732FCAB91D57c;2;2;A833467508CFD705c,9277683CC735A247c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_01ABEE1192B6A70Fa;2;2;A4DD1617B00EFB3Aa,DED95D621B123E75a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_01ABEE1192B6A70Fb;2;2;A4DD1617B00EFB3Ab,DED95D621B123E75b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_01ABEE1192B6A70Fc;2;2;A4DD1617B00EFB3Ac,DED95D621B123E75c,

-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D0AEBE42A78C2504a;2;2;601F9B511EFCF222a,BCD071335241B21Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D0AEBE42A78C2504b;2;2;601F9B511EFCF222b,BCD071335241B21Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D0AEBE42A78C2504c;2;2;601F9B511EFCF222c,BCD071335241B21Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_312EBA581A706047a;2;2;AA45C33E4BB7ED70a,8896E405CE9E4105a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_312EBA581A706047b;2;2;AA45C33E4BB7ED70b,8896E405CE9E4105b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_312EBA581A706047c;2;2;AA45C33E4BB7ED70c,8896E405CE9E4105c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_3033C26D4640CF67a;2;2;A917EA31DAB5817Da,709D5530F4FC4B0Da,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_3033C26D4640CF67b;2;2;A917EA31DAB5817Db,709D5530F4FC4B0Db,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_3033C26D4640CF67c;2;2;A917EA31DAB5817Dc,709D5530F4FC4B0Dc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_553C8611F8CEA90Fa;2;2;EE685B3E0CF8B329a,40C3F902EEB91E56a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_553C8611F8CEA90Fb;2;2;EE685B3E0CF8B329b,40C3F902EEB91E56b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_553C8611F8CEA90Fc;2;2;EE685B3E0CF8B329c,40C3F902EEB91E56c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0EECE54C847DBD66a;2;2;373F5D2751838D1Da,E2B6016E76F4B965a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0EECE54C847DBD66b;2;2;373F5D2751838D1Db,E2B6016E76F4B965b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0EECE54C847DBD66c;2;2;373F5D2751838D1Dc,E2B6016E76F4B965c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_116671273A694654a;2;2;117696009762282Ea,80251E2EC66C7B63a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_116671273A694654b;2;2;117696009762282Eb,80251E2EC66C7B63b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_116671273A694654c;2;2;117696009762282Ec,80251E2EC66C7B63c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CD62D247A27C0B6Ca;2;2;EC00C139B3F76151a,88F9DF5723AA6A6Ca,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CD62D247A27C0B6Cb;2;2;EC00C139B3F76151b,88F9DF5723AA6A6Cb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CD62D247A27C0B6Cc;2;2;EC00C139B3F76151c,88F9DF5723AA6A6Cc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C1A1BD4E4C4FFF2Ea;2;2;1E10C745D6204208a,558F3365A17F1C63a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C1A1BD4E4C4FFF2Eb;2;2;1E10C745D6204208b,558F3365A17F1C63b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_C1A1BD4E4C4FFF2Ec;2;2;1E10C745D6204208c,558F3365A17F1C63c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0BF98245ED4A2975a;2;2;F30E3B38FAC8C151a,A9E68607328AB530a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0BF98245ED4A2975b;2;2;F30E3B38FAC8C151b,A9E68607328AB530b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_0BF98245ED4A2975c;2;2;F30E3B38FAC8C151c,A9E68607328AB530c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_17A9DC7616D36459a;2;2;97745117104DB150a,B54FBF48D898A61Aa,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_17A9DC7616D36459b;2;2;97745117104DB150b,B54FBF48D898A61Ab,
1E+15,1E+15,0,0,0,,.1,.001,5,

_SW0;_17A9DC7616D36459C;2;2;97745117104DB150c,B54FBF48D898A61Ac,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_8A006416273AED2Fa;2;2;F8DBFD62BCDF465Da,26000F5A3B659626a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_8A006416273AED2Fb;2;2;F8DBFD62BCDF465Db,26000F5A3B659626b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_8A006416273AED2Fc;2;2;F8DBFD62BCDF465Dc,26000F5A3B659626c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_BC3FA0271C5ECB1Aa;2;2;80728238CB219121a,371E5474FDFA7C2Aa,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_BC3FA0271C5ECB1Ab;2;2;80728238CB219121b,371E5474FDFA7C2Ab,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_BC3FA0271C5ECB1Ac;2;2;80728238CB219121c,371E5474FDFA7C2Ac,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CF301F1D26958B43a;2;2;5B468D4607139D69a,A28ED34FBCAE5963a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_CF301F1D26958B43b;2;2;5B468D4607139D69b,A28ED34FBCAE5963b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_CF301F1D26958B43c;2;2;5B468D4607139D69c,A28ED34FBCAE5963c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_65B5A44B28FB204Ca;2;2;FBD7D244124CFA59a,671CB9228068F230a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_65B5A44B28FB204Cb;2;2;FBD7D244124CFA59b,671CB9228068F230b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_65B5A44B28FB204Cc;2;2;FBD7D244124CFA59c,671CB9228068F230c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D66E5620806ABE4Da;2;2;CE50263754C4B422a,9E0ECD16EC5EE41Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D66E5620806ABE4Db;2;2;CE50263754C4B422b,9E0ECD16EC5EE41Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_D66E5620806ABE4Dc;2;2;CE50263754C4B422c,9E0ECD16EC5EE41Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_016DF90258DE517Ca;2;2;8D2ABB22040DD35Da,C6CECD0F73BF8D63a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_016DF90258DE517Cb;2;2;8D2ABB22040DD35Db,C6CECD0F73BF8D63b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_016DF90258DE517Cc;2;2;8D2ABB22040DD35Dc,C6CECD0F73BF8D63c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CDE00C015EFB594Aa;2;2;7B5BF00DCB79E36Fa,E468D313586A5077a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CDE00C015EFB594Ab;2;2;7B5BF00DCB79E36Fb,E468D313586A5077b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_CDE00C015EFB594Ac;2;2;7B5BF00DCB79E36Fc,E468D313586A5077c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_97E60F0B39CDE819a;2;2;C2B3562506AFEF45a,6BCED8475F1D0371a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_97E60F0B39CDE819b;2;2;C2B3562506AFEF45b,6BCED8475F1D0371b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_97E60F0B39CDE819c;2;2;C2B3562506AFEF45c,6BCED8475F1D0371c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_9164A54938B3A426a;2;2;80D55556AB7EC176a,40294F6B2276B32Ba,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9164A54938B3A426b;2;2;80D55556AB7EC176b,40294F6B2276B32Bb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9164A54938B3A426c;2;2;80D55556AB7EC176c,40294F6B2276B32Bc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_87508C3BD1759C3Aa;2;2;C954B91953924742a,0BD8B02E0C91C62Fa,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_87508C3BD1759C3Ab;2;2;C954B91953924742b,0BD8B02E0C91C62Fb,

-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_87508C3BD1759C3AC;2;2;C954B91953924742C,0BD8B02E0C91C62Fc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9AB42D665B46AA17a;2;2;0BD8B02E0C91C62Fa,EFD8A053826AB11Fa,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9AB42D665B46AA17b;2;2;0BD8B02E0C91C62Fb,EFD8A053826AB11Fb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_9AB42D665B46AA17c;2;2;0BD8B02E0C91C62Fc,EFD8A053826AB11Fc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_4AF076529FAE3231a;2;2;92971A19E7FD9967a,A9E68607328AB530a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_4AF076529FAE3231b;2;2;92971A19E7FD9967b,A9E68607328AB530b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_4AF076529FAE3231c;2;2;92971A19E7FD9967c,A9E68607328AB530c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E31C472E44C90906a;2;2;0277827F8A023F3Aa,C423EE218A59BC29a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E31C472E44C90906b;2;2;0277827F8A023F3Ab,C423EE218A59BC29b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_E31C472E44C90906c;2;2;0277827F8A023F3Ac,C423EE218A59BC29c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_A931A65EF65F1E69a;2;2;97D48338817CE76Ea,8896E405CE9E4105a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_A931A65EF65F1E69b;2;2;97D48338817CE76Eb,8896E405CE9E4105b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_A931A65EF65F1E69c;2;2;97D48338817CE76Ec,8896E405CE9E4105c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_9FE99042C452AC5Fa;2;2;DA9B155D9FA61123a,8896E405CE9E4105a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_9FE99042C452AC5Fb;2;2;DA9B155D9FA61123b,8896E405CE9E4105b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_9FE99042C452AC5Fc;2;2;DA9B155D9FA61123c,8896E405CE9E4105c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_7E30FD6078AC7D3Ca;2;2;FE8B2715FEE0F112a,1E10C745D6204208a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_7E30FD6078AC7D3Cb;2;2;FE8B2715FEE0F112b,1E10C745D6204208b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_7E30FD6078AC7D3Cc;2;2;FE8B2715FEE0F112c,1E10C745D6204208c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_78434D42F9A9B363a;2;2;C423EE218A59BC29a,B64802328E7DF765a,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_78434D42F9A9B363b;2;2;C423EE218A59BC29b,B64802328E7DF765b,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_78434D42F9A9B363c;2;2;C423EE218A59BC29c,B64802328E7DF765c,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_8D0B520605CF6826a;2;2;73E6EC01EBD66710a,13136855F5863B5Ca,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_8D0B520605CF6826b;2;2;73E6EC01EBD66710b,13136855F5863B5Cb,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_8D0B520605CF6826c;2;2;73E6EC01EBD66710c,13136855F5863B5Cc,
-1,1E+15,0,0,0,,.1,.001,5,
_SW0;_90614920CE911832a;2;2;BE80427AE64E0E78a,28E74E02E0530E12a,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_90614920CE911832b;2;2;BE80427AE64E0E78b,28E74E02E0530E12b,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SW0;_90614920CE911832c;2;2;BE80427AE64E0E78c,28E74E02E0530E12c,
1E+15,1E+15,0,0,0,,.1,.001,5,
_SM;_96298B26B2FB9743a;3;1;40CD876D36D0B974a,
_SM;_96298B26B2FB9743b;3;1;40CD876D36D0B974b,

[illegible]

_PQload; _66B16602EB971255c;1;1; B7389473D674782Cc,
138kVRMSLL,1.44469165802002MW 0 15.0993633270264MVar 0 60 1 0
_PQload; _4FD93A35E3D09615a;1;1; 7093D25AF7428B35a,
345kVRMSLL,156.206085205078MW 0 29.6854724884033MVar 0 60 1 0
_PQload; _4FD93A35E3D09615b;1;1; 7093D25AF7428B35b,
345kVRMSLL,156.206085205078MW 0 29.6854724884033MVar 0 60 1 0
_PQload; _4FD93A35E3D09615c;1;1; 7093D25AF7428B35c,
345kVRMSLL,156.206085205078MW 0 29.6854724884033MVar 0 60 1 0
_PQload; _F910204347750A3Fa;1;1; 0818100980654841a,
13.8000001907349kVRMSLL,4.82835102081299MW 0 1.65453553199768MVar 0 60 1 0
_PQload; _F910204347750A3Fb;1;1; 0818100980654841b,
13.8000001907349kVRMSLL,4.82835102081299MW 0 1.65453553199768MVar 0 60 1 0
_PQload; _F910204347750A3Fc;1;1; 0818100980654841c,
13.8000001907349kVRMSLL,4.82835102081299MW 0 1.65453553199768MVar 0 60 1 0
_PQload; _BB67407BF9EA7D29a;1;1; 2F56BA6B50A4D74Ba,
345kVRMSLL,149.709594726563MW 0 14.1830797195435MVar 0 60 1 0
_PQload; _BB67407BF9EA7D29b;1;1; 2F56BA6B50A4D74Bb,
345kVRMSLL,149.709594726563MW 0 14.1830797195435MVar 0 60 1 0
_PQload; _BB67407BF9EA7D29c;1;1; 2F56BA6B50A4D74Bc,
345kVRMSLL,149.709594726563MW 0 14.1830797195435MVar 0 60 1 0
_PQload; _F404814E0AF7C914a;1;1; 97D48338817CE76Ea,
22kVRMSLL,89.5927276611328MW 0 20.4183902740479MVar 0 60 1 0
_PQload; _F404814E0AF7C914b;1;1; 97D48338817CE76Eb,
22kVRMSLL,89.5927276611328MW 0 20.4183902740479MVar 0 60 1 0
_PQload; _F404814E0AF7C914c;1;1; 97D48338817CE76Ec,
22kVRMSLL,89.5927276611328MW 0 20.4183902740479MVar 0 60 1 0
_PQload; _AA1C4A044A212721a;1;1; 3232311F943C7523a,
13.8000001907349kVRMSLL,28.1538162231445MW 0 4.63596868515015MVar 0 60 1 0
_PQload; _AA1C4A044A212721b;1;1; 3232311F943C7523b,
13.8000001907349kVRMSLL,28.1538162231445MW 0 4.63596868515015MVar 0 60 1 0
_PQload; _AA1C4A044A212721c;1;1; 3232311F943C7523c,
13.8000001907349kVRMSLL,28.1538162231445MW 0 4.63596868515015MVar 0 60 1 0
_PQload; _DF85673839C04643a;1;1; A833467508CFD705a,
22kVRMSLL,72.6300888061523MW 0 47.7117767333984MVar 0 60 1 0
_PQload; _DF85673839C04643b;1;1; A833467508CFD705b,
22kVRMSLL,72.6300888061523MW 0 47.7117767333984MVar 0 60 1 0
_PQload; _DF85673839C04643c;1;1; A833467508CFD705c,
22kVRMSLL,72.6300888061523MW 0 47.7117767333984MVar 0 60 1 0
_PQload; _7CC6F23DDCD91169a;1;1; 1EA0A91354C96133a,
138kVRMSLL,277.809020996094MW 0 61.986270904541MVar 0 60 1 0
_PQload; _7CC6F23DDCD91169b;1;1; 1EA0A91354C96133b,
138kVRMSLL,277.809020996094MW 0 61.986270904541MVar 0 60 1 0
_PQload; _7CC6F23DDCD91169c;1;1; 1EA0A91354C96133c,
138kVRMSLL,277.809020996094MW 0 61.986270904541MVar 0 60 1 0
_PQload; _1B22221E9677E517a;1;1; C9C8B84BBF2BC705a,
13.8000001907349kVRMSLL,5.28031015396118MW 0 3.52020692825317MVar 0 60 1 0
_PQload; _1B22221E9677E517b;1;1; C9C8B84BBF2BC705b,
13.8000001907349kVRMSLL,5.28031015396118MW 0 3.52020692825317MVar 0 60 1 0
_PQload; _1B22221E9677E517c;1;1; C9C8B84BBF2BC705c,
13.8000001907349kVRMSLL,5.28031015396118MW 0 3.52020692825317MVar 0 60 1 0
_PQload; _763CE43EFFF01615a;1;1; EDD8134417161A29a,
345kVRMSLL,121.348403930664MW 0 35.8296279907227MVar 0 60 1 0
_PQload; _763CE43EFFF01615b;1;1; EDD8134417161A29b,
345kVRMSLL,121.348403930664MW 0 35.8296279907227MVar 0 60 1 0
_PQload; _763CE43EFFF01615c;1;1; EDD8134417161A29c,
345kVRMSLL,121.348403930664MW 0 35.8296279907227MVar 0 60 1 0
_PQload; _80C18F06BBC28A23a;1;1; 13C79967ABDE7C57a,
345kVRMSLL,582.984008789063MW 0 131.992721557617MVar 0 60 1 0
_PQload; _80C18F06BBC28A23b;1;1; 13C79967ABDE7C57b,

```

345kVRMSLL,582.984008789063MW 0 131.992721557617MVar 0 60 1 0
_PQload; _80C18F06BBC28A23c;1;1; 13C79967ABDE7C57c,
345kVRMSLL,582.984008789063MW 0 131.992721557617MVar 0 60 1 0
_PQload; _AFBFD40D8AC14747a;1;1; A6B3CC0D52A13804a,
345kVRMSLL,36.5797576904297MW 0 4.82319641113281MVar 0 60 1 0
_PQload; _AFBFD40D8AC14747b;1;1; A6B3CC0D52A13804b,
345kVRMSLL,36.5797576904297MW 0 4.82319641113281MVar 0 60 1 0
_PQload; _AFBFD40D8AC14747c;1;1; A6B3CC0D52A13804c,
345kVRMSLL,36.5797576904297MW 0 4.82319641113281MVar 0 60 1 0
_PQload; _D7CBAF7BA3535A7Da;1;1; 70551608D7F26555a,
13.8000001907349kVRMSLL,116.414215087891MW 0 18.1990337371826MVar 0 60 1 0
_PQload; _D7CBAF7BA3535A7Db;1;1; 70551608D7F26555b,
13.8000001907349kVRMSLL,116.414215087891MW 0 18.1990337371826MVar 0 60 1 0
_PQload; _D7CBAF7BA3535A7Dc;1;1; 70551608D7F26555c,
13.8000001907349kVRMSLL,116.414215087891MW 0 18.1990337371826MVar 0 60 1 0
_PQload; _CCE2FC785D0B8466a;1;1; D3C212544CE75437a,
22kVRMSLL,52.3817329406738MW 0 7.83423948287964MVar 0 60 1 0
_PQload; _CCE2FC785D0B8466b;1;1; D3C212544CE75437b,
22kVRMSLL,52.3817329406738MW 0 7.83423948287964MVar 0 60 1 0
_PQload; _CCE2FC785D0B8466c;1;1; D3C212544CE75437c,
22kVRMSLL,52.3817329406738MW 0 7.83423948287964MVar 0 60 1 0
_PQload; _85CBF50815366860a;1;1; F010C746427EAD4Aa,
13.8000001907349kVRMSLL,92.4553146362305MW 0 13.0757732391357MVar 0 60 1 0
_PQload; _85CBF50815366860b;1;1; F010C746427EAD4Ab,
13.8000001907349kVRMSLL,92.4553146362305MW 0 13.0757732391357MVar 0 60 1 0
_PQload; _85CBF50815366860c;1;1; F010C746427EAD4Ac,
13.8000001907349kVRMSLL,92.4553146362305MW 0 13.0757732391357MVar 0 60 1 0
_PQload; _30E0207F5BC7131Ea;1;1; 27591F2A01142231a,
22kVRMSLL,177.307769775391MW 0 61.972339630127MVar 0 60 1 0
_PQload; _30E0207F5BC7131Eb;1;1; 27591F2A01142231b,
22kVRMSLL,177.307769775391MW 0 61.972339630127MVar 0 60 1 0
_PQload; _30E0207F5BC7131Ec;1;1; 27591F2A01142231c,
22kVRMSLL,177.307769775391MW 0 61.972339630127MVar 0 60 1 0
_PQload; _562D5C0707B45A56a;1;1; 977F7655EC1A022Ea,
22kVRMSLL,68.0869445800781MW 0 17.8489379882813MVar 0 60 1 0
_PQload; _562D5C0707B45A56b;1;1; 977F7655EC1A022Eb,
22kVRMSLL,68.0869445800781MW 0 17.8489379882813MVar 0 60 1 0
_PQload; _562D5C0707B45A56c;1;1; 977F7655EC1A022Ec,
22kVRMSLL,68.0869445800781MW 0 17.8489379882813MVar 0 60 1 0
_PQload; _AE26EE024B07DD6Aa;1;1; DA9B155D9FA61123a,
22kVRMSLL,87.6256484985352MW 0 39.9234199523926MVar 0 60 1 0
_PQload; _AE26EE024B07DD6Ab;1;1; DA9B155D9FA61123b,
22kVRMSLL,87.6256484985352MW 0 39.9234199523926MVar 0 60 1 0
_PQload; _AE26EE024B07DD6Ac;1;1; DA9B155D9FA61123c,
22kVRMSLL,87.6256484985352MW 0 39.9234199523926MVar 0 60 1 0
_PQload; _BA61F70C56B42E43a;1;1; 30546F43BD0F8533a,
345kVRMSLL,318.402984619141MW 0 2.41109347343445MVar 0 60 1 0
_PQload; _BA61F70C56B42E43b;1;1; 30546F43BD0F8533b,
345kVRMSLL,318.402984619141MW 0 2.41109347343445MVar 0 60 1 0
_PQload; _BA61F70C56B42E43c;1;1; 30546F43BD0F8533c,
345kVRMSLL,318.402984619141MW 0 2.41109347343445MVar 0 60 1 0
_PQload; _3F556E7C08BDF849a;1;1; C1C7F47F1682D105a,
138kVRMSLL,2.93598699569702MW 0 30.245325088501MVar 0 60 1 0
_PQload; _3F556E7C08BDF849b;1;1; C1C7F47F1682D105b,
138kVRMSLL,2.93598699569702MW 0 30.245325088501MVar 0 60 1 0
_PQload; _3F556E7C08BDF849c;1;1; C1C7F47F1682D105c,
138kVRMSLL,2.93598699569702MW 0 30.245325088501MVar 0 60 1 0
_PQload; _3DF490442F99C82Da;1;1; 4FFF1B064FF0B60Aa,
13.8000001907349kVRMSLL,4.75350379943848MW 0 2.37675285339355MVar 0 60 1 0

```

```

_PQload; _3DF490442F99C82Db;1;1; 4FFF1B064FF0B60Ab,
13.8000001907349kVRMSLL,4.75350379943848MW 0 2.37675285339355MVar 0 60 1 0
_PQload; _3DF490442F99C82Dc;1;1; 4FFF1B064FF0B60Ac,
13.8000001907349kVRMSLL,4.75350379943848MW 0 2.37675285339355MVar 0 60 1 0
_PQload; _06DE213E80D1EE48a;1;1; 5040590A34818C37a,
138kVRMSLL,410.471008300781MW 0 186.557434082031MVar 0 60 1 0
_PQload; _06DE213E80D1EE48b;1;1; 5040590A34818C37b,
138kVRMSLL,410.471008300781MW 0 186.557434082031MVar 0 60 1 0
_PQload; _06DE213E80D1EE48c;1;1; 5040590A34818C37c,
138kVRMSLL,410.471008300781MW 0 186.557434082031MVar 0 60 1 0
_PQload; _152B367CADF5D502a;1;1; 7148F27A5B75E40Ea,
13.8000001907349kVRMSLL,5.27440118789673MW 0 3.5162672996521MVar 0 60 1 0
_PQload; _152B367CADF5D502b;1;1; 7148F27A5B75E40Eb,
13.8000001907349kVRMSLL,5.27440118789673MW 0 3.5162672996521MVar 0 60 1 0
_PQload; _152B367CADF5D502c;1;1; 7148F27A5B75E40Ec,
13.8000001907349kVRMSLL,5.27440118789673MW 0 3.5162672996521MVar 0 60 1 0
_PQload; _096A383B1431FF4Aa;1;1; A5FC1D6E9BA8E600a,
22kVRMSLL,45.5386810302734MW 0 9.07874584197998MVar 0 60 1 0
_PQload; _096A383B1431FF4Ab;1;1; A5FC1D6E9BA8E600b,
22kVRMSLL,45.5386810302734MW 0 9.07874584197998MVar 0 60 1 0
_PQload; _096A383B1431FF4Ac;1;1; A5FC1D6E9BA8E600c,
22kVRMSLL,45.5386810302734MW 0 9.07874584197998MVar 0 60 1 0
_PQload; _E48C3C22DCA7FB24a;1;1; CE50263754C4B422a,
345kVRMSLL,325.645385742188MW 0 31.9209842681885MVar 0 60 1 0
_PQload; _E48C3C22DCA7FB24b;1;1; CE50263754C4B422b,
345kVRMSLL,325.645385742188MW 0 31.9209842681885MVar 0 60 1 0
_PQload; _E48C3C22DCA7FB24c;1;1; CE50263754C4B422c,
345kVRMSLL,325.645385742188MW 0 31.9209842681885MVar 0 60 1 0
_PQload; _A2CBA93CD90FA449a;1;1; 827EFF2D454DFE29a,
13.8000001907349kVRMSLL,105.881950378418MW 0 26.5365047454834MVar 0 60 1 0
_PQload; _A2CBA93CD90FA449b;1;1; 827EFF2D454DFE29b,
13.8000001907349kVRMSLL,105.881950378418MW 0 26.5365047454834MVar 0 60 1 0
_PQload; _A2CBA93CD90FA449c;1;1; 827EFF2D454DFE29c,
13.8000001907349kVRMSLL,105.881950378418MW 0 26.5365047454834MVar 0 60 1 0
<xmfr_YY_unitO_c8188d0a;4;i,j,k,m,
_RLC;RL1;2;2;i,s36,
#R1#,#L1#,0,0,0,
#W1_scope#,
_RLC;RL2;2;2;s31,k,
#R2#,#L2#,0,0,0,
#W2_scope#,
_Tr0;Trs;4;4;s36,j,s31,m
#Ratio#,,
_x_Lnonl;Lmag;2;2;s36,j,
1,#Phi0#,1e-08,
#Lmag_scope#,
#ILnonl# #PhiLnonl#
_x_Rmag;Rmag;2;2;s36,j,
#Rm#,,,,
>
<YY_O_3lee8dcd;8;Ya,Yb,Yc,Youta,Youtb,Youtc,NW1,NW2,
@xmfr_YY_unitO_c8188d0a;xfmr_A;4;Ya,NW1,Youta,NW2,
Phi0=Phiss01
@xmfr_YY_unitO_c8188d0a;xfmr_B;4;Yb,NW1,Youtb,NW2,
Phi0=Phiss02
@xmfr_YY_unitO_c8188d0a;xfmr_C;4;Yc,NW1,Youtc,NW2,
Phi0=Phiss03
>
!
```

```

!
@YY_O_31ee8dcd;_814A01103192BB7E;8;B2D87B3EFC54347Da,B2D87B3EFC54347Db,B2D87B3E
FC54347Dc,DED95D621B123E75a,DED95D621B123E75b,DED95D621B123E75c,N1,N2
R1=1.9044;
L1=51.775874;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.4;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_CD73D4690805A368;8;B15662314D1B733Fa,B15662314D1B733Fb,B1566231
4D1B733Fc,ED96D66420E88D15a,ED96D66420E88D15b,ED96D66420E88D15c,N1,N2
R1=0;
L1=34.041149;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.0400000005528548;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_8E32431BCE364F1A;8;1F6A1678C541CF05a,1F6A1678C541CF05b,1F6A1678
C541CF05c,9CD76D6BE17F0105a,9CD76D6BE17F0105b,9CD76D6BE17F0105c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=28.566001;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=1.725;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_20FF292437EE5225;8;0646C66D2980A347a,0646C66D2980A347b,0646C66D
2980A347c,CF6C3F4DF2AD1749a,CF6C3F4DF2AD1749b,CF6C3F4DF2AD1749c,N1,N2
R1=0;
L1=0;

```

```

Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0.9522;
L2=18.567899;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=15.6818181818182;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_5C9FA021DCEDCA10;8;B5FAFF7E2A61B94Fa,B5FAFF7E2A61B94Fb,B5FAFF7E
2A61B94Fc,8B54113CB432E22Ba,8B54113CB432E22Bb,8B54113CB432E22Bc,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0.9522;
L2=23.3289;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=24.9999996544658;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_2674BB112AB64E67;8;53797B5EBC7CB439a,53797B5EBC7CB439b,53797B5E
BC7CB439c,0CEB5532D83A4051a,0CEB5532D83A4051b,0CEB5532D83A4051c,N1,N2
R1=0.9522;
L1=22.3767;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.4;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_78E77D0D488D8F5D;8;DEE8F26AFE0A4021a,DEE8F26AFE0A4021b,DEE8F26A
FE0A4021c,DA4827344EED162Ea,DA4827344EED162Eb,DA4827344EED162Ec,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';

```

```

R2=0.9522;
L2=25.114274;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=2.5;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_84D726017EFDDA08;8;117696009762282Ea,117696009762282Eb,11769600
9762282Ec,616275244D7CAE77a,616275244D7CAE77b,616275244D7CAE77c,N1,N2
R1=0;
L1=34.041149;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.0400000005528548;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_8290581AEFC28E4A;8;BCB0F348D164C41Ca,BCB0F348D164C41Cb,BCB0F348
D164C41Cc,A80A65213C7B5751a,A80A65213C7B5751b,A80A65213C7B5751c,N1,N2
R1=0.892688;
L1=23.924025;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.4;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_EB6AB06C3628FD6A;8;077D32076FB90731a,077D32076FB90731b,077D3207
6FB90731c,97745117104DB150a,97745117104DB150b,97745117104DB150c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=4.8E-05;
L2=0.0484;
Phiss01=0;
Phiss02=0;
Phiss03=0;

```

```

Ratio=22;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_87E678385C6CC222;8;F2B2301D76C1346Ca,F2B2301D76C1346Cb,F2B2301D
76C1346Cc,87C27121ED1EEF38a,87C27121ED1EEF38b,87C27121ED1EEF38c,N1,N2
R1=0;
L1=34.041149;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.0400000005528548;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_BDA18162847FBF37;8;B506C20669F69D21a,B506C20669F69D21b,B506C206
69F69D21c,1293C157C664DF3Ba,1293C157C664DF3Bb,1293C157C664DF3Bc,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=28.566001;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=1.725;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_3D6EE1142F35987E;8;F45D493F2C04250Ca,F45D493F2C04250Cb,F45D493F
2C04250Cc,58F6211107750854a,58F6211107750854b,58F6211107750854c,N1,N2
R1=0.892688;
L1=23.924025;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.4;
Lmag_scope='';
W1_scope='';
W2_scope='';

```

```

@YY_O_31ee8dcd;_9C28D316E7E48778;8;4613006609E16B4Da,4613006609E16B4Db,46130066
09E16B4Dc,884C6A372F530007a,884C6A372F530007b,884C6A372F530007c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=34.041149;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=24.9999996544658;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_1253321F8FC50200;8;D3A7F4077D662C72a,D3A7F4077D662C72b,D3A7F407
7D662C72c,98AF910BECCC8E45a,98AF910BECCC8E45b,98AF910BECCC8E45c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=1.9E-05;
L2=0.019044;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=13.8000001907349;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_97753464CD035E7F;8;FAB63908248C8C38a,FAB63908248C8C38b,FAB63908
248C8C38c,FEBFC76E4BB0951Aa,FEBFC76E4BB0951Ab,FEBFC76E4BB0951Ac,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0.133308;
L2=3.865932;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=6.27272727272727;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_FCBFB55E054A9226;8;5EF6976722F3E36Ba,5EF6976722F3E36Bb,5EF69767
22F3E36Bc,BDFF711011376A4Da,BDFF711011376A4Db,BDFF711011376A4Dc,N1,N2
R1=0;
L1=34.041149;
Rm=0;

```

```

PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.0400000005528548;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_FDACCA7E5461962E;8;671CB9228068F230a,671CB9228068F230b,671CB922
8068F230c,C6C1406A3194A954a,C6C1406A3194A954b,C6C1406A3194A954c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0.71415;
L2=20.9484;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=15.6818181818182;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_D247BB5E9051CB4D;8;0B4AA268392D925Aa,0B4AA268392D925Ab,0B4AA268
392D925Ac,1448C1713BA1374Ba,1448C1713BA1374Bb,1448C1713BA1374Bc,N1,N2
R1=0.171396;
L1=3.789756;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.159420289855072;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_4A13221371290846;8;2F6DCC3CEF8BF569a,2F6DCC3CEF8BF569b,2F6DCC3C
EF8BF569c,3A962615EA528071a,3A962615EA528071b,3A962615EA528071c,N1,N2
R1=0.71415;
L1=27.613799;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;

```

```

L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.0400000005528548;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_AA7870163DD49434;8;E9ED6E22D47DAC49a,E9ED6E22D47DAC49b,E9ED6E22
D47DAC49c,77C3AB49608B6828a,77C3AB49608B6828b,77C3AB49608B6828c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=34.041149;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=24.9999996544658;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_F07DE252DFEFC3B;8;59E4567D6A66CE6Da,59E4567D6A66CE6Db,59E4567D
6A66CE6Dc,15A39B340D9ECA0Ca,15A39B340D9ECA0Cb,15A39B340D9ECA0Cc,N1,N2
R1=0;
L1=34.041149;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.0400000005528548;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_B47EF72BF1D1B34F;8;31CC367C1FE9183Ea,31CC367C1FE9183Eb,31CC367C
1FE9183Ec,C50F745C1E57492Ca,C50F745C1E57492Cb,C50F745C1E57492Cc,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0.595125;
L2=32.3748;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=15.6818181818182;

```

```

Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_DCDFD844A461685A;8;DC2FF6414E0E5D47a,DC2FF6414E0E5D47b,DC2FF641
4E0E5D47c,E468D313586A5077a,E468D313586A5077b,E468D313586A5077c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=1.9044;
L2=51.775874;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=2.5;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_84AD8D1C9DB3521D;8;BB52462116E49328a,BB52462116E49328b,BB524621
16E49328c,6E323109C3F7FA54a,6E323109C3F7FA54b,6E323109C3F7FA54c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0.9522;
L2=25.114274;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=2.5;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_3011797FC4AC7A04;8;40CD876D36D0B974a,40CD876D36D0B974b,40CD876D
36D0B974c,7FB36165489E8C75a,7FB36165489E8C75b,7FB36165489E8C75c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=34.041149;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=24.9999996544658;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_340CF81440687371;8;F9DFFA5932869516a,F9DFFA5932869516b,F9DFFA59
32869516c,DC84A430732F4F47a,DC84A430732F4F47b,DC84A430732F4F47c,N1,N2

```

```

R1=0;
L1=23.805;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.0400000005528548;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_9D6AF1491A40AD56;8;26000F5A3B659626a,26000F5A3B659626b,26000F5A
3B659626c,B5CD6D28E8821A22a,B5CD6D28E8821A22b,B5CD6D28E8821A22c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=11.9025;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=1;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_42200C5223F8EF1E;8;183A2A16AA261713a,183A2A16AA261713b,183A2A16
AA261713c,D5596E4146DE113Fa,D5596E4146DE113Fb,D5596E4146DE113Fc,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=4.8E-05;
L2=0.0484;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=22;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_DC7004091DF26A5B;8;C203F07798607476a,C203F07798607476b,C203F077
98607476c,8277354589523D57a,8277354589523D57b,8277354589523D57c,N1,N2
R1=0;
L1=34.041149;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];

```

```

Lnnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.0400000005528548;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_DF50F069D52E4723;8;BFF38270DBF83039a,BFF38270DBF83039b,BFF38270
DBF83039c,BD6EDA06FF1CD976a,BD6EDA06FF1CD976b,BD6EDA06FF1CD976c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnnonl='Exclude';
Rmag='Exclude';
R2=0.595125;
L2=19.75815;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=24.9999996544658;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_901C4D5467947832;8;EB583F48D7EDBF0Aa,EB583F48D7EDBF0Ab,EB583F48
D7EDBF0Ac,7346E11A6B5C9309a,7346E11A6B5C9309b,7346E11A6B5C9309c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=34.041149;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=24.9999996544658;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_28BF2604B6106B51;8;2F0A2247C2CBC128a,2F0A2247C2CBC128b,2F0A2247
C2CBC128c,FDA7FC5780D5427Ea,FDA7FC5780D5427Eb,FDA7FC5780D5427Ec,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnnonl='Exclude';
Rmag='Exclude';
R2=0.9522;
L2=22.3767;
Phiss01=0;

```

```

Phiss02=0;
Phiss03=0;
Ratio=2.5;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_8818211CBCDFD720;8;4BFA5623F5D89002a,4BFA5623F5D89002b,4BFA5623
F5D89002c,E81B402AB89F3F61a,E81B402AB89F3F61b,E81B402AB89F3F61c,N1,N2
R1=0.833175;
L1=22.257675;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.0400000005528548;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_1D1C1E53BA490156;8;D2D640537EB38F0Aa,D2D640537EB38F0Ab,D2D64053
7EB38F0Ac,294E2F711CD69F74a,294E2F711CD69F74b,294E2F711CD69F74c,N1,N2
R1=0.171396;
L1=3.42792;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.100000001382137;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_A47A904E3F961E13;8;0D2B461A6822DE2Da,0D2B461A6822DE2Db,0D2B461A
6822DE2Dc,87507B47331D932Ba,87507B47331D932Bb,87507B47331D932Bc,N1,N2
R1=0;
L1=29.75625;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.0400000005528548;
Lmag_scope='';
W1_scope='';

```

```

W2_scope = '';
@YY_O_31ee8dcd;_B935A230BD0AFC59;8;E5934D31D9EDCB62a,E5934D31D9EDCB62b,E5934D31
D9EDCB62c,C389D6640FD79F2Fa,C389D6640FD79F2Fb,C389D6640FD79F2Fc,N1,N2
R1=0.171396;
L1=3.42792;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.100000001382137;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_7E648B61F5F51B08;8;5A66AD6903F3667Da,5A66AD6903F3667Db,5A66AD69
03F3667Dc,4C87317B0036E835a,4C87317B0036E835b,4C87317B0036E835c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0.833175;
L2=23.447925;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=15.6818181818182;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_61DF1C61B24E1621;8;85E55D69ABE5930Fa,85E55D69ABE5930Fb,85E55D69
ABE5930Fc,F2750720CF356304a,F2750720CF356304b,F2750720CF356304c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0.833175;
L2=16.42545;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=2.5;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_19319E04300D4901;8;D3A7F4077D662C72a,D3A7F4077D662C72b,D3A7F407
7D662C72c,EB998F2991147947a,EB998F2991147947b,EB998F2991147947c,N1,N2
R1=0;
L1=0;

```

```

Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=4.8E-05;
L2=0.0484;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=22;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_C4ACCD193112004F;8;0F0CE67017A3A379a,0F0CE67017A3A379b,0F0CE670
17A3A379c,183A2A16AA261713a,183A2A16AA261713b,183A2A16AA261713c,N1,N2
R1=0.595125;
L1=21.18645;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.00289855072463768;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_1775F739DEA90C5E;8;D04D655C8E21AF4Fa,D04D655C8E21AF4Fb,D04D655C
8E21AF4Fc,43F8C5234DAC1507a,43F8C5234DAC1507b,43F8C5234DAC1507c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=21.543526;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=24.9999996544658;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_F9237D553E06C82D;8;193F33563A4F710Ea,193F33563A4F710Eb,193F3356
3A4F710Ec,077D32076FB90731a,077D32076FB90731b,077D32076FB90731c,N1,N2
R1=0.09522;
L1=3.389832;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';

```

```

R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.0072463768115942;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_FDD29269E3F80C04;8;E3DDBF26AE79550Ca,E3DDBF26AE79550Cb,E3DDBF26
AE79550Cc,6A37E80B57F1F456a,6A37E80B57F1F456b,6A37E80B57F1F456c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0.171396;
L2=3.42792;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=9.99999986178631;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_E339C241B6873134;8;F70E5A3F25746026a,F70E5A3F25746026b,F70E5A3F
25746026c,077D32076FB90731a,077D32076FB90731b,077D32076FB90731c,N1,N2
R1=1.9E-05;
L1=0.019044;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.0724637671143935;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_4E2F6A14B87DE215;8;863D8457254C3B16a,863D8457254C3B16b,863D8457
254C3B16c,183A2A16AA261713a,183A2A16AA261713b,183A2A16AA261713c,N1,N2
R1=1.9E-05;
L1=0.019044;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;

```

```

Ratio=0.0714285714285714;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_F66F6211E4060C61;8;BECEC67082E25F0Da,BECEC67082E25F0Db,BECEC670
82E25F0Dc,2FF83207E206145Ba,2FF83207E206145Bb,2FF83207E206145Bc,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=28.566001;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=1.725;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_65107239F4BE1822;8;80728238CB219121a,80728238CB219121b,80728238
CB219121c,ADB4F13B1E391241a,ADB4F13B1E391241b,ADB4F13B1E391241c,N1,N2
R1=0.833175;
L1=22.257675;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.0400000005528548;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_A0DC965D8FEA0A3F;8;A26EC97EE927216Fa,A26EC97EE927216Fb,A26EC97E
E927216Fc,90E0C050B3CBE74Ea,90E0C050B3CBE74Eb,90E0C050B3CBE74Ec,N1,N2
R1=0.833175;
L1=16.901549;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=0;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=0.063768115942029;
Lmag_scope='';
W1_scope='';
W2_scope='';

```

```

@YY_O_31ee8dcd;_C29A8448573E521A;8;704E7A481933725Ea,704E7A481933725Eb,704E7A48
1933725Ec,4BC2A53215E5D162a,4BC2A53215E5D162b,4BC2A53215E5D162c,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=34.041149;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=24.9999996544658;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_29FA93129EF5060A;8;B323A4660DF0FD52a,B323A4660DF0FD52b,B323A466
0DF0FD52c,2C1383656D1E417Aa,2C1383656D1E417Ab,2C1383656D1E417Ac,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0.9522;
L2=23.3289;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=24.9999996544658;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_D0A3BC5DBA400B40;8;E92BB83F26A37E16a,E92BB83F26A37E16b,E92BB83F
26A37E16c,AC0D116F0F141E3Aa,AC0D116F0F141E3Ab,AC0D116F0F141E3Ac,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0.171396;
L2=3.42792;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=9.99999986178631;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_FC56F060894A3979;8;D3A7F4077D662C72a,D3A7F4077D662C72b,D3A7F407
7D662C72c,C192583DEFC58611a,C192583DEFC58611b,C192583DEFC58611c,N1,N2
R1=0;
L1=0;
Rm=0;

```

```

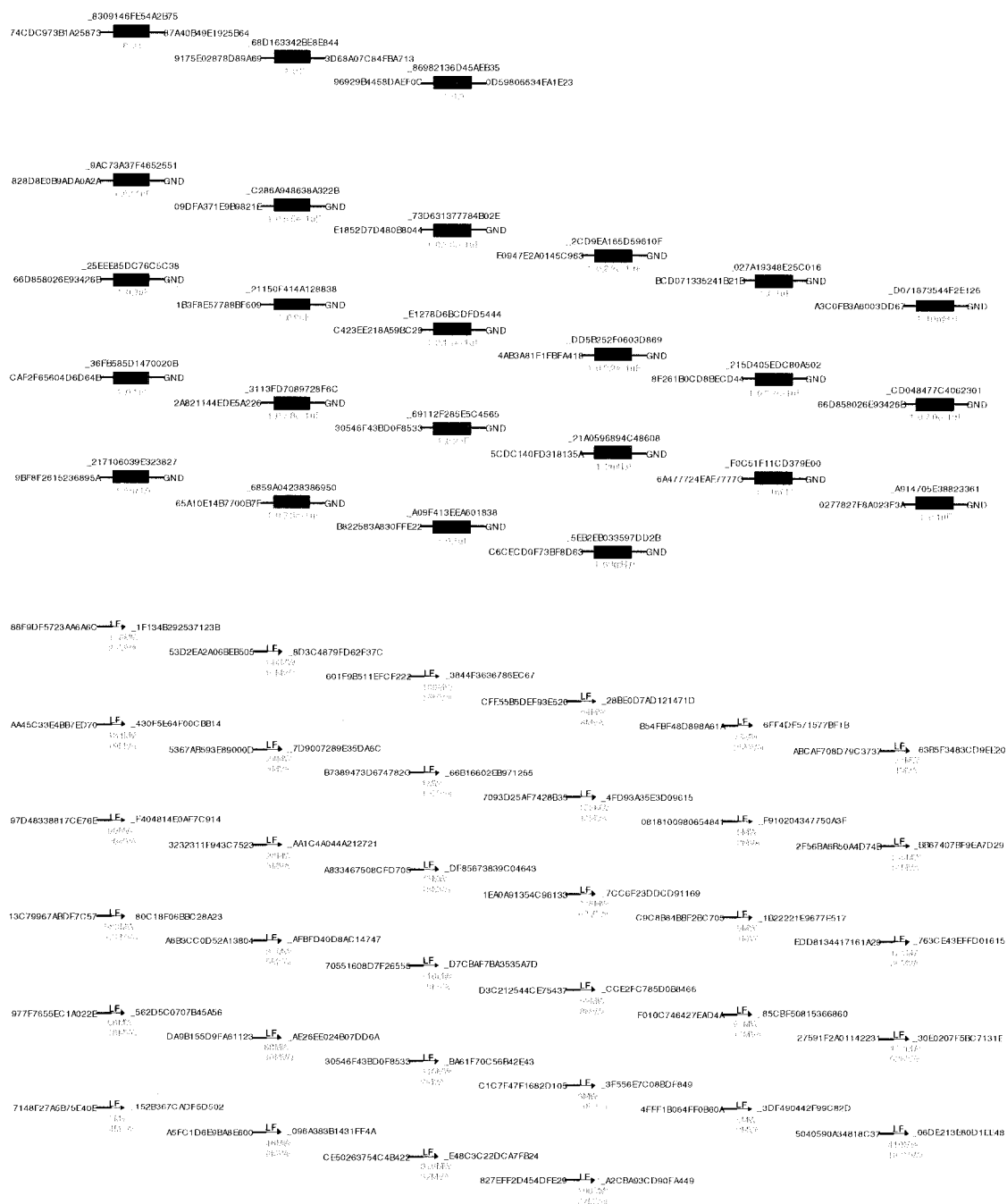
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0.09522;
L2=3.389832;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=138;
Lmag_scope='';
W1_scope='';
W2_scope='';
@YY_O_31ee8dcd;_37958C402F7AD436;8;B7935647EF65833Da,B7935647EF65833Db,B7935647
EF65833Dc,581948073631FB1Da,581948073631FB1Db,581948073631FB1Dc,N1,N2
R1=0;
L1=0;
Rm=0;
PhiLnonl=[0];
ILnonl=[0];
Lnonl='Exclude';
Rmag='Exclude';
R2=0;
L2=28.566001;
Phiss01=0;
Phiss02=0;
Phiss03=0;
Ratio=1.725;
Lmag_scope='';
W1_scope='';
W2_scope='';
_RLC;_CHE-CHF2;2;2;74CDC973B1A25873,87A40B49E1925B64,
8.30000019073486,0,0,0,0,0,
_RLC;_CHE-CHF1;2;2;9175E02878D89A69,3D68A07C84FBA713,
3.84999990463257,0,0,0,0,0,
_RLC;_CHE-CHF3;2;2;96929B4458DAEF0C,0D59806634EA1E23,
3.84999990463257,0,0,0,0,0,
_RLC;_9AC73A37F4652551a;1;1;828D8E0B9ADA0A2Aa,
,,11.1429631794368uF,,0,
_RLC;_9AC73A37F4652551b;1;1;828D8E0B9ADA0A2Ab,
,,11.1429631794368uF,,0,
_RLC;_9AC73A37F4652551c;1;1;828D8E0B9ADA0A2Ac,
,,11.1429631794368uF,,0,
_RLC;_C286A948638A322Ba;1;1;09DFA371E9B9821Ea,
,,0.546005578355682uF,,0,
_RLC;_C286A948638A322Bb;1;1;09DFA371E9B9821Eb,
,,0.546005578355682uF,,0,
_RLC;_C286A948638A322Bc;1;1;09DFA371E9B9821Ec,
,,0.546005578355682uF,,0,
_RLC;_73D631377784B02Ea;1;1;E1852D7D480B8044a,
,,0.334288895383103uF,,0,
_RLC;_73D631377784B02Eb;1;1;E1852D7D480B8044b,
,,0.334288895383103uF,,0,
_RLC;_73D631377784B02Ec;1;1;E1852D7D480B8044c,
,,0.334288895383103uF,,0,
_RLC;_2CD9EA165D59610Fa;1;1;E0947E2A0145C963a,
,,0.222859263588735uF,,0,
_RLC;_2CD9EA165D59610Fb;1;1;E0947E2A0145C963b,
,,0.222859263588735uF,,0,

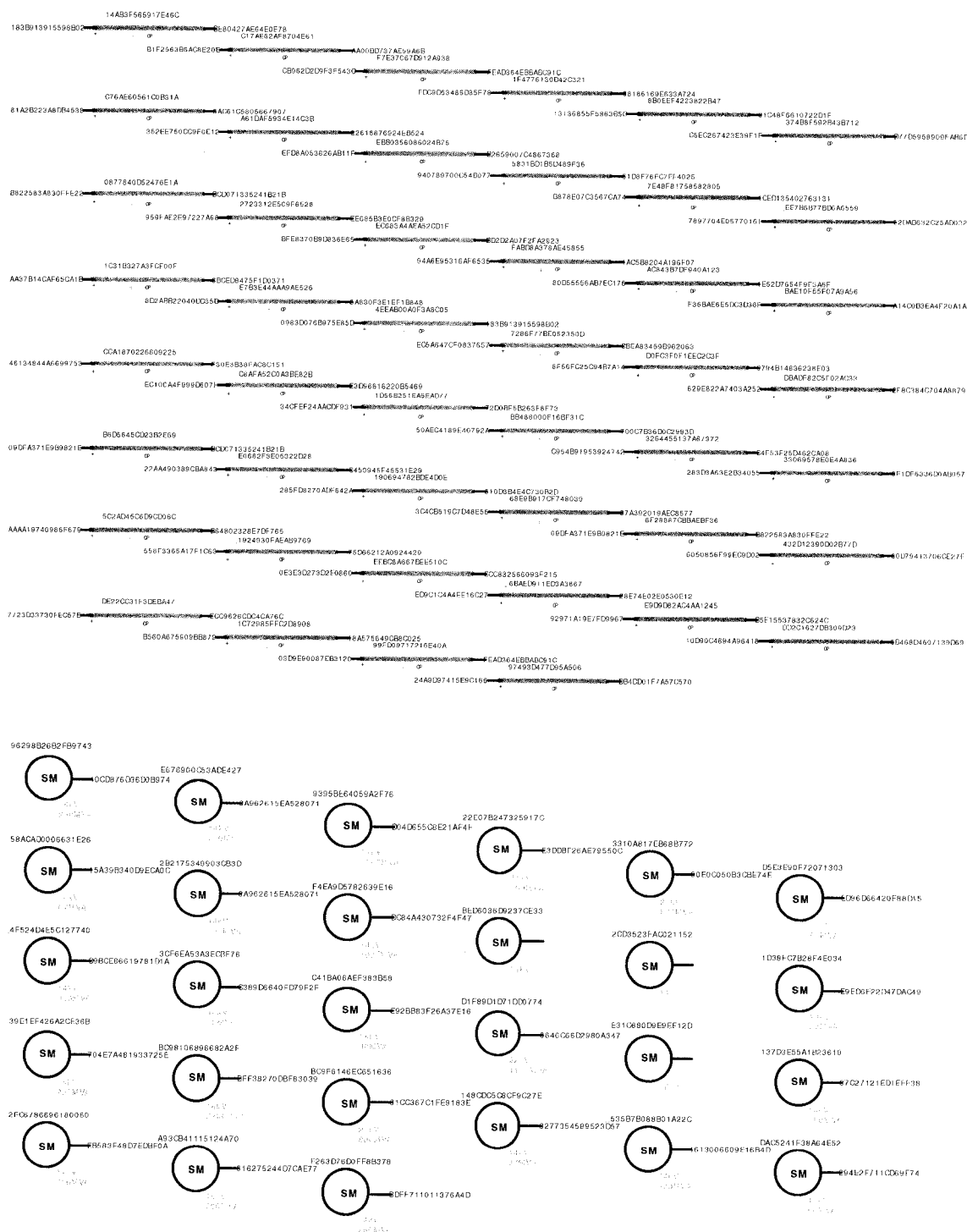
```

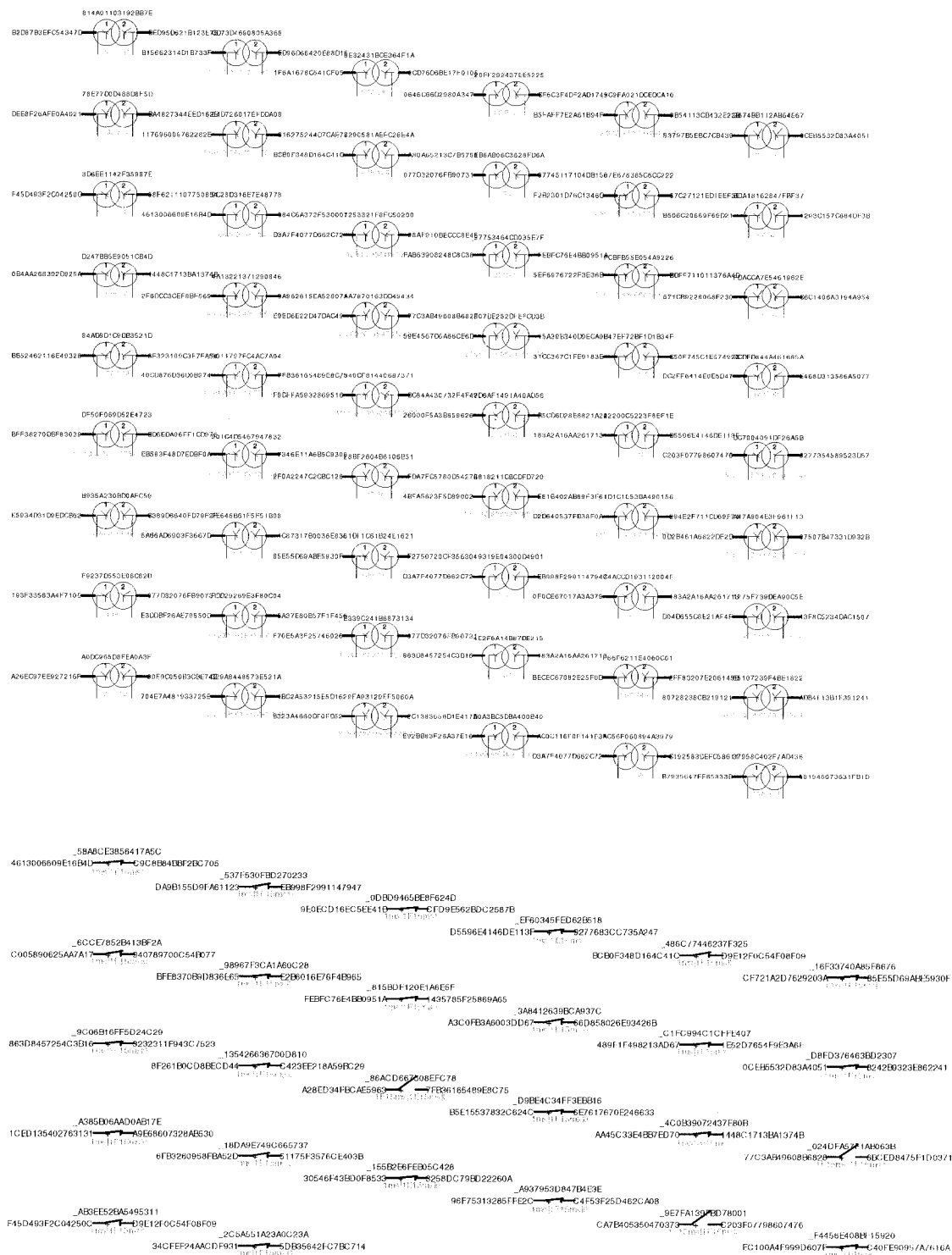
_RLC;_2CD9EA165D59610Fc;1;1;E0947E2A0145C963c,
,,0.222859263588735uF,,0,
_RLC;_027A19348E25C016a;1;1;BCD071335241B21Ba,
,,1.42407540411107uF,,0,
_RLC;_027A19348E25C016b;1;1;BCD071335241B21Bb,
,,1.42407540411107uF,,0,
_RLC;_027A19348E25C016c;1;1;BCD071335241B21Bc,
,,1.42407540411107uF,,0,
_RLC;_D071873544F2E126a;1;1;A3C0FB3A6003DD67a,
,,10.1031557874735,,0,,
_RLC;_D071873544F2E126b;1;1;A3C0FB3A6003DD67b,
,,10.1031557874735,,0,,
_RLC;_D071873544F2E126c;1;1;A3C0FB3A6003DD67c,
,,10.1031557874735,,0,,
_RLC;_25EEE85DC76C5C38a;1;1;66D858026E93426Ba,
,,2.78574079485919uF,,0,
_RLC;_25EEE85DC76C5C38b;1;1;66D858026E93426Bb,
,,2.78574079485919uF,,0,
_RLC;_25EEE85DC76C5C38c;1;1;66D858026E93426Bc,
,,2.78574079485919uF,,0,
_RLC;_21150F414A128838a;1;1;1B3F8E57788BF609a,
,,9.05365758329237uF,,0,
_RLC;_21150F414A128838b;1;1;1B3F8E57788BF609b,
,,9.05365758329237uF,,0,
_RLC;_21150F414A128838c;1;1;1B3F8E57788BF609c,
,,9.05365758329237uF,,0,
_RLC;_E1278D6BCDFD5444a;1;1;C423EE218A59BC29a,
,,0.111429631794368uF,,0,
_RLC;_E1278D6BCDFD5444b;1;1;C423EE218A59BC29b,
,,0.111429631794368uF,,0,
_RLC;_E1278D6BCDFD5444c;1;1;C423EE218A59BC29c,
,,0.111429631794368uF,,0,
_RLC;_DD5B252F0603D869a;1;1;4AB3A81F1FBFA418a,
,,0.222859263588735uF,,0,
_RLC;_DD5B252F0603D869b;1;1;4AB3A81F1FBFA418b,
,,0.222859263588735uF,,0,
_RLC;_DD5B252F0603D869c;1;1;4AB3A81F1FBFA418c,
,,0.222859263588735uF,,0,
_RLC;_215D405EDC80A502a;1;1;8F261B0CD8BECD44a,
,,0.668577790766206uF,,0,
_RLC;_215D405EDC80A502b;1;1;8F261B0CD8BECD44b,
,,0.668577790766206uF,,0,
_RLC;_215D405EDC80A502c;1;1;8F261B0CD8BECD44c,
,,0.668577790766206uF,,0,
_RLC;_CD048477C4062301a;1;1;66D858026E93426Ba,
,,0.696435198714798uF,,0,
_RLC;_CD048477C4062301b;1;1;66D858026E93426Bb,
,,0.696435198714798uF,,0,
_RLC;_CD048477C4062301c;1;1;66D858026E93426Bc,
,,0.696435198714798uF,,0,
_RLC;_36FB585D1470020Ba;1;1;CAF2F65604D6D64Ba,
,,1.11429631794368uF,,0,
_RLC;_36FB585D1470020Bb;1;1;CAF2F65604D6D64Bb,
,,1.11429631794368uF,,0,
_RLC;_36FB585D1470020Bc;1;1;CAF2F65604D6D64Bc,
,,1.11429631794368uF,,0,
_RLC;_3113FD7089728F6Ca;1;1;2A821144EDE5A226a,
,,0.780007422560574uF,,0,
_RLC;_3113FD7089728F6Cb;1;1;2A821144EDE5A226b,

,,0.780007422560574uF,,0,
_RLC;_3113FD7089728F6CC;1;1;2A821144EDE5A226C,
,,0.780007422560574uF,,0,
_RLC;_69112F285E5C4565a;1;1;30546F43BD0F8533a,
,,1.78287410870988uF,,0,
_RLC;_69112F285E5C4565b;1;1;30546F43BD0F8533b,
,,1.78287410870988uF,,0,
_RLC;_69112F285E5C4565c;1;1;30546F43BD0F8533c,
,,1.78287410870988uF,,0,
_RLC;_21A0596894C48608a;1;1;5CDC140FD318135Aa,
,1.68385929791225,,0,,
_RLC;_21A0596894C48608b;1;1;5CDC140FD318135Ab,
,1.68385929791225,,0,,
_RLC;_21A0596894C48608c;1;1;5CDC140FD318135Ac,
,1.68385929791225,,0,,
_RLC;_F0C51F11CD379E00a;1;1;6A477724EAE7777Ca,
,10.5241206119516,,0,,
_RLC;_F0C51F11CD379E00b;1;1;6A477724EAE7777Cb,
,10.5241206119516,,0,,
_RLC;_F0C51F11CD379E00c;1;1;6A477724EAE7777Cc,
,10.5241206119516,,0,,
_RLC;_A914705E38823361a;1;1;0277827F8A023F3Aa,
,,1.11429631794368uF,,0,
_RLC;_A914705E38823361b;1;1;0277827F8A023F3Ab,
,,1.11429631794368uF,,0,
_RLC;_A914705E38823361c;1;1;0277827F8A023F3Ac,
,,1.11429631794368uF,,0,
_RLC;_217106039E323827a;1;1;9BF8F2615236895Aa,
,5.26206030597579,,0,,
_RLC;_217106039E323827b;1;1;9BF8F2615236895Ab,
,5.26206030597579,,0,,
_RLC;_217106039E323827c;1;1;9BF8F2615236895Ac,
,5.26206030597579,,0,,
_RLC;_6859A04238386950a;1;1;65A10E14B7700B7Fa,
,,0.334288895383103uF,,0,
_RLC;_6859A04238386950b;1;1;65A10E14B7700B7Fb,
,,0.334288895383103uF,,0,
_RLC;_6859A04238386950c;1;1;65A10E14B7700B7Fc,
,,0.334288895383103uF,,0,
_RLC;_A09F413EEA601838a;1;1;B822583A830FFE22a,
,,1.41404570007802uF,,0,
_RLC;_A09F413EEA601838b;1;1;B822583A830FFE22b,
,,1.41404570007802uF,,0,
_RLC;_A09F413EEA601838c;1;1;B822583A830FFE22c,
,,1.41404570007802uF,,0,
_RLC;_5EB2EB033597DD2Ba;1;1;C6CECD0F73BF8D63a,
,63.1447236717095,,0,,
_RLC;_5EB2EB033597DD2Bb;1;1;C6CECD0F73BF8D63b,
,63.1447236717095,,0,,
_RLC;_5EB2EB033597DD2Bc;1;1;C6CECD0F73BF8D63c,
,63.1447236717095,,0,,

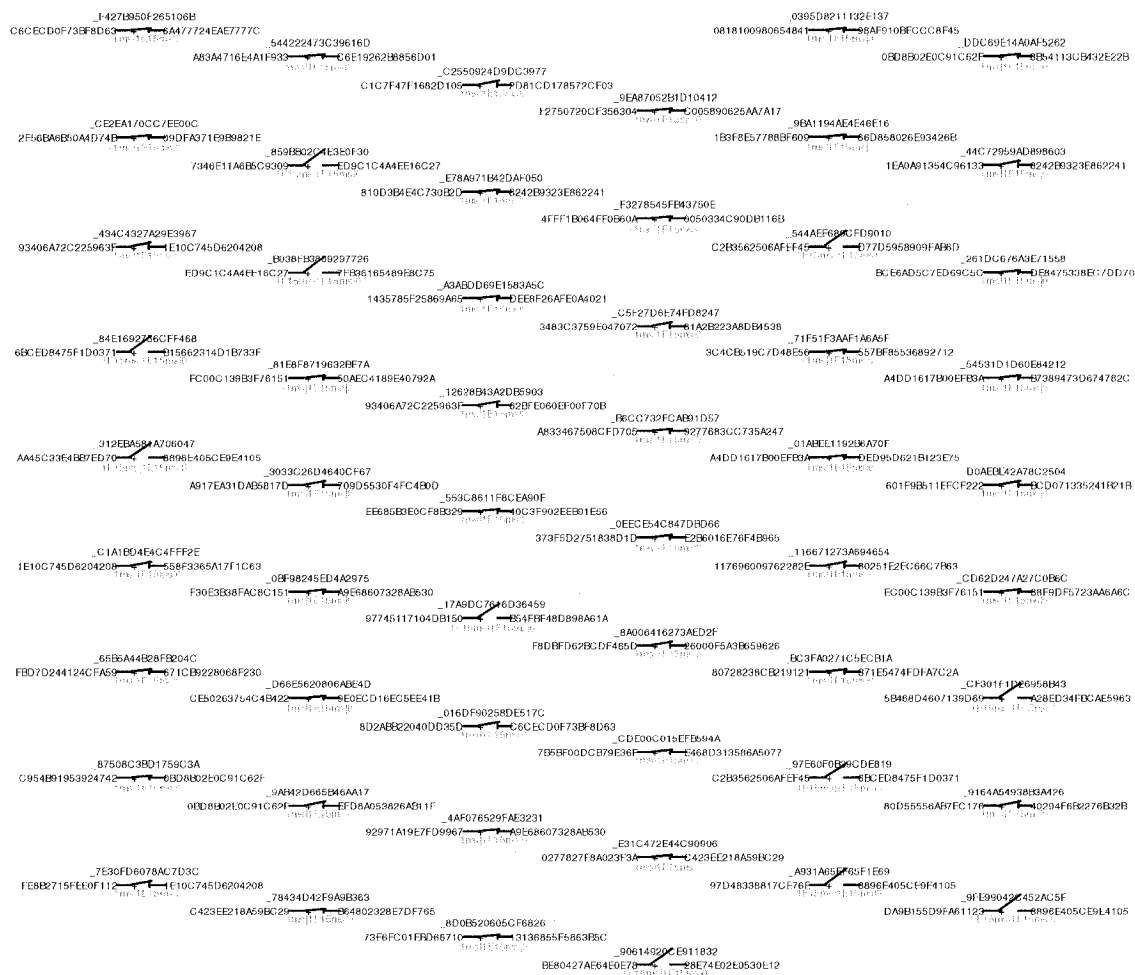
Solution frequency: 60











ANNEXE H : L'annuaire UDDI

UDDI (Universal Description Discovery and Integration) est une norme industrielle, édictée par OASIS³¹, pour la publication et la localisation d'informations relatives aux services Web. Il s'agit d'un annuaire mondial qui permet :

- de publier la description d'un service Web
- de découvrir les services Web disponibles pour les applications clientes.

Plus précisément, UDDI définit :

- un **modèle de données** permettant de décrire un service Web. Les données sont écrites en XML et le modèle se présente sous forme de schéma XML.
- des **interfaces** pour publier les descriptions des services dans le registre et pour interroger le registre afin de chercher des descriptions publiées.

Parce que les interfaces UDDI sont aussi spécifiées en WSDL avec une attache SOAP (accès aux données par messages SOAP), le registre peut être invoqué par la communauté UDDI comme un service Web.

1. Modèle de données

UDDI est un annuaire orienté *Business*. Le noyau d'UDDI travaille avec la notion de "business registry", qui est un service sophistiqué de noms et répertoires.

Il décrit quatre types d'entités :

- les "BusinessEntities" décrivent les organisations ayant publié des services dans le répertoire. Les informations sur l'organisation sont notamment fournies : nom, coordonnées, liste des contacts, ...
- les "ServiceEntities" : décrivent de manière non technique les services proposés par les différentes organisations. Ils comportent notamment le nom et la description

³¹ OASIS (*Organization for the Advancement of Structured Information Standards*) est un consortium mondial qui travaille pour la normalisation et la standardisation de formats de fichiers ouverts basés notamment sur XML. Il a été créé en 1993 et compte 3500 membres faisant partie de 600 organisations.

textuelle des services ainsi qu'une référence à l'organisation proposant le service et à un ou plusieurs "bindingTemplates".

- les "technical models" (tModels) : ce sont les descriptions techniques des services. WSDL est alors choisi comme le vocabulaire pour publier des descriptions techniques de services.
- les "BindingTemplates" donnent les coordonnées des services. Ils contiennent notamment une description, la définition du point d'accès du service et les éventuels tModels associés.

L'annuaire UDDI est consultable de différentes manières :

- Les **pages blanches** comprennent la liste des entreprises (BusinessEntities)
- Les **pages jaunes** recensent les services Web (ServiceEntity) de chacune des entreprises sous le standard WSDL.
- Les **pages vertes** fournissent des informations complémentaires ainsi que le mode d'emploi de l'annuaire

2. Interface UDDI

L'interface UDDI est définie sous forme de document UDDI et implémentée sous forme de service Web SOAP. Elle est composée de plusieurs modules, entre autres :

- **Interrogation** ("*Inquiry*") : interface permettant de rechercher des informations dans un répertoire UDDI et de lire les différents enregistrements suivant le modèle de données UDDI.
- **Publication** : interface permettant de publier des informations dans un répertoire UDDI conformément à son modèle de données.
- **Contrôle d'accès et propriété** ("*Custody and Ownership transfer*") : interface permettant de transférer la propriété d'informations et de gérer les droits d'accès associés.

3. Enregistrement d'un service

Ce processus comporte quatre étapes :

1. couper en deux le fichier WSDL de description du service proposé, en séparant la partie abstraite de la partie concrète.
2. enregistrer la partie abstraite comme un tModel
3. s'enregistrer comme une Business Entity
4. enregistrer un serviceEntity associant la BusinessEntity avec le tModel.

Remarques :

- tout est décrit en XML
- le tModel fait référence à la partie abstraite du WSDL. Le BindingTemplate pourrait faire référence à la partie concrète.
- chaque élément de l'annuaire est identifié par un UUID (Universal Unique ID). Les UUID sont utilisés pour lier les éléments entre eux.

Pour plus d'information sur UDDI, consulter son document de spécification à l'adresse :

<http://www.uddi.org/specification.html>

ANNEXE I : Méthodes et propriétés pour l'objet XMLHttpRequest

1. PROPRIETES

- ***readyState***: représente l'état d'avancement de la requête (en lecture seule). La propriété ***readyState*** prend successivement 5 valeurs récapitulées dans le tableau suivant :

Valeur	Signification	Description
0	Uninitialized	Non initialisé
1	Loading	Début du transfert des données
2	Loaded	Données transférées
3	Interactive	Les données reçues sont accessibles en partie
4	complete	Les données sont complètement accessibles

- ***onreadystatechange*** spécifie la fonction à appeler lorsque la propriété ***readyState*** varie (lecture/écriture).
- ***responseText*** : Chaîne de caractères contenant la réponse à la requête (en lecture seule).
- ***responseXML*** : Objet XML contenant la réponse à la requête (lecture seule).
- ***status*** : représente le code HTTP retourné par la requête (en lecture seule). Le tableau ci-dessous liste la valeur des différents codes ainsi que leur signification.

Code	Description	Code	Description
100	Continue	404	Not Found
101	Switching Protocols	405	Method Not Allowed
200	OK	406	Not Acceptable
201	Created	407	Proxy Authentication Required
202	Accepted	408	Request Timeout
203	Non-Authoritative Information	409	Conflict
204	No Content	410	Gone
205	Reset Content	411	Length Required
206	Partial Content	412	Precondition Failed
300	Multiple Choices	413	Request Entity too large
301	Moved Permanently	414	Request-URI too long
302	Found	415	Unsupported Media Type
303	See Other	416	Requested Range Not Suitable
304	Not Modified	417	Expectation Failed
305	Use Proxy	500	Internal Server Error
307	Temporary Redirect	501	Not Implemented
400	Bad Request	502	Bad Gateway
401	Unauthorized	503	Service Unavailable
402	Payment Required	504	Gateway Timeout
403	Forbidden	505	HTTP Version Not Supported

2. METHODES

- **Abort ()** → annule la requête courante
- **getAllresponseHeaders ()** → retourne les noms et les valeurs de tous les en-têtes http sous la forme d'une chaîne de caractères
- **getResponseHeader (headerName)** → récupère la valeur d'un certain en-tête HTTP (headerName) sous forme d'une chaîne de caractères
- **setRequestHeader (headerName, headerValue)** → spécifie un en-tête http *headerName* et *headerValue* à envoyer avec la requête.
- **open (method, url, asynchrone, user, password)** → initialise la requête en spécifiant la methode (*method*), l'URL (*url*), si le mode est asynchrone ou non (c'est un booléen : *true* ou *false*) et en indiquant d'éventuelles informations d'identification (*user* et *password*).
- **Send (data)** : envoie la requête HTTP au serveur en transmettant éventuellement des données. Data doit alors être différent de *null*.

Le lien suivant fournit une liste et une description plus exhaustive de toutes les méthodes et propriétés des objets XMLHttpRequest :

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/xmlsdk/html/1aac5bb8-3647-44cf-a1ce-c220ba464509.asp>

ANNEXE I : Résumé des espaces de noms utilisés

Préfixe	URI de l'espace de noms	Description
cim	http://iec.ch/TC57/2000/CIM-schema-cim10#	Espace de noms pour les objets CIM
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	Espace de noms pour les éléments RDF
rdfs	http://www.w3.org/2000/01/rdf-schema#	RDF Schema
soap	http://schemas.xmlsoap.org/wsdl/soap/	Espace de noms WSDL pour l'association ("binding") WSDL SOAP
soapenc	http://schemas.xmlsoap.org/soap/encoding/	Espace de noms pour l'encodage défini par SOAP 1.1 (cf. référence [16])
soapenv	http://schemas.xmlsoap.org/soap/envelope/	Espace de noms pour l'enveloppe défini par SOAP 1.1 (cf. référence [16])
xsd	http://www.w3.org/2000/10/XMLSchema	Espace de noms pour le schéma XSD
wSDL	http://schemas.xmlsoap.org/wsdl/	Espace de noms pour WSDL

ANNEXE J : Liste des fichiers du CD-ROM

Nom du fichier	Description
cimComplet.cs	Code des classes de types CIM et RDF
Program.cs	Code pour la génération du Netlist
cim2emtp.cs	Code contenant les méthodes exposées par le serveur COM
test_cim.dwj	Application JavaScript utilisant les méthodes exposées par le serveur COM pour importer les données CIM-XML vers EMTP-RV
Service.cs	Code contenant les méthodes Web exposées par le serveur Web
test_Web.dwj	Application JavaScript utilisant les méthodes Web pour importer les données CIM-XML vers EMTP-RV
cim.xml	Document CIM-XML du modèle de réseau à 60 barres
res_PM.xml	Document CIM-XML du modèle de réseau à 6 barres

**Un cédérom accompagne ce
mémoire de maîtrise**

**Toute personne intéressée à se le
procurer doit contacter :**

**An accompanying CD-ROM is
included with the original thesis.**

**To obtain a copy of the CD-ROM,
please contact :**

École Polytechnique de Montréal

Service du prêt entre bibliothèques

B.P. 6079, Succursale Centre-Ville

Montréal, Québec H3C 3A7

Canada

Tél. : (514) 340-4846

Télec./Fax : (514) 340-4026