N° d'ordre 669

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir le grade de

DOCTEUR D'UNIVERSITE
spécialité productique

par

LUC COUTEREEL
Maître en informatique
Mastère IDN

ETUDE ET REALISATION D'UN PROCESSEUR D'AIDE A LA MODELISATION DES SYSTEMES DYNAMIQUES PAR L'APPROCHE BOND-GRAPH.

soutenue le 22 Janvier 1991 devant le jury d'examen

Mr C.ROMBAUT	président
Mr J.C. GENTINA	rapporteur
Mr M.LEBRUN	rapporteur
Mme G.DAUPHIN TANGUY	examinateur
Mr J.P.RICHARD	examinateur
Mr M.STAROSWIECKI	examinateur

Thèse dirigée par **Mme G.DAUPHIN TANGUY,** Professeur à l'IDN.

Je dédie cette thèse

aux Miens

et à tous ceux qui me sont Chers .

AVANT - PROPOS

Avant-propos

Ce travail de recherche a été effectué au laboratoire d'Automatique et d'Informatique Industrielle de l'IDN, sous la direction de Madame G .DAUPHIN TANGUY,Professeur à l'IDN.

Nous sommes trés reconnaissant à Madame G.DAUPHIN TANGUY pour nous avoir soutenu tout au long de ces trois années.

Et ceci quelles que soient les difficultés, qu'elle trouve ici notre profonde amitié pour sa bienveillance et sa gentillesse.

Nous remercions Monsieur C.ROMBAUT, Professeur à l'IDN de nous honorer en présidant cette thèse.

Nous remercions de même Monsieur J.C.GENTINA, Professeur et Directeur de l'IDN et Monsieur M.LEBRUN, Professeur à l'université de Lyon pour l'attention qu'ils ont portée à l'évolution de nos travaux.

Nous remercions aussi Monsieur J.P.RICHARD, Professeur à l'IDN et Monsieur M.STAROSWIECKY, Professeur à l'université de Lille I pour leur participation au jury de thèse.

Nous adressons aux professeurs aux chercheurs et au personnel de l'IDN toute notre sympathie pour l'accueil chaleureux qu'ils nous ont réservé au sein du laboratoire.

TABLE DES MATIERES .

TABLE DES MATIERES .

Chapitre I L'outil , les Bond-Graphs .

I) Introduction
II) Représentation des transferts de puissance
III) Variables mises en jeu
III-1) Variables de puissance
III-2) Variables d'énergie
IV) Les éléments Bond-Graphs
IV-1) Eléments actifs : les sources
IV-2) Les éléments passifs : R , C , I
a) Elément dissipatif d'énergie
b) Les éléments de stockage d'énergie
IV-3) Les éléments de jonction
a) Les éléments transducteurs
b) Les éléments jonction-0 et jonction-1
IV-4) Tableau récapitulatif
V) Procédure de construction d'un modèle Bond-Graph
V-1) Procédure pour modéliser par Bond-Graph un système mécanique en dimension 1
V-2) Procédure pour modéliser par Bond-Graph un système électrique
VI) Notion de causalité
VI-1) Les causalités obligatoires
VI-2) Restrictions de causalité
VI-3) Les causalités préférentielles
VI-4) Procédure d'affectation de la causalité
VII) Détermination de l'équation d'état
VII-1) Variables d'état
VII-2)Méthode générale de mise en équation
VIII) Conclusion

Chapitre II Le contexte informatique .

I) Introduction

II) Logiciels utilisant l'approche Bond-Graph
- II-1) Enport
- II-2) Tutsim
- II-3) Camp
- II-4) Scribt
- II-5) Sidops
- II-6) Simodo
- II-7) Spécificité d'Archer

III) Intelligence artificielle , système expert
- III-1) Présentation générale d'un système expert
- III-2) Fonctionnement d'un système expert
- III-3) Organisation d'un système expert

IV) Présentation générale d'Archer
- IV-1) Historique
- IV-2) Organisation d'Archer
  - IV-2-1) Base de connaissances
  - IV-2-2) Un moteur d'inférences
  - IV-2-3) Langage d'expression des connaissances
  - IV-2-4) Organisation fonctionnelle en modules

V) Conclusion

Chapitre III Le système expert Archer

I) Introduction

II) Base de faits : "présentation des modèles en langage naturel"

II-1) Modèle mécanique en dimension 1

II-2) Modèle électrique

II-3) Modèle Bond-Graph

II-4) Modèle équation

II-5) Evolution des langages descriptifs

III) Base de règles : "les algorithmes"

III-1) Construction du Bond-Graph non simplifié et sans causalité à partir d'un modèle mécanique de dimension 1

III-2) Construction du Bond-Graph non simplifié et sans causalité à partir d'un modèle électrique

III-3) Simplification du Bond-Graph

III-4) Placement des causalités

III-5) Construction de l'équation d'état associée au modèle

IV) Le superviseur : "les menus"

V) Conclusion

Chapitre IV

Spécifications de la partie couplage .

I) Introduction

II) Modèles de couplage

II-1) Modèles à coupler

II-2) Modèles de système de couplage

II-3) Editeurs de couplage

III) Algorithmes de couplage

IV) Exemple

V) Conclusion

INTRODUCTION GENERALE .

Introduction générale

Les travaux présentés dans ce mémoire ont été réalisés au laboratoire d'Automatique et d'Informatique industrielle de l'IDN, sous la direction de Madame le Professeur Geneviève Dauphin-Tanguy .

Ils concernent l'étude et la réalisation d'un processeur d'aide à la modélisation nommé Archer utilisant conjointement les outils de l'intelligence Artificielle et la technique Bond-Graph.

Les Bond-Graphs, introduits par Paynter [Paynter1961] et développés par Karnopp,Rosenberg [Karnopp,Rosenberg 1975], [Rosenber, Karnopp1983] Thoma [Thoma 1975] ,Breedveld [breedveld 1979] constituent un outil graphique de représentation des transferts de puissance au sein d'un système physique.

Avec un langage unique, quel que soit le domaine physique concerné, il est possible de transcrire graphiquement toutes les informations énergétiques , et les relations de cause-à-effet, correspondant aux phénomènes physiques mis en jeu.

Le modèle Bond-Graph apparait comme un intermédiaire trés performant entre le schéma physique du système étudié et ses modèles mathématiques associés (équation d'état, fonction ou matrice de transfert, équations différentielles du second ordre),linéaires ou non linéaires.

C'est également un outil d'analyse trés puissant , qui permet de mettre en évidence des propriétés structurelles du système (commandabilité,observabillité,stabilité asymptotique...).

Cependant,malgré les grands apports de cet outil, il est indéniable qu'une phase d'apprentissage incontournable du langage Bond-Graph peut apparaître fastidieuse aux utilisateurs potentiels.

Ceci explique peut être pourquoi l'approche Bond-Graph est encore peu développée , et reste actuellement le fait de quelques spécialistes,même si l'on rencontre depuis quelques années une curiosité grandissante chez les scientifiques.

La finalité du projet Archer est donc de permettre à un utilisateur de bénéficier des potentialités de l'outil Bond-Graph,tout en le dispensant de cette phase d'apprentissage.

L'approche Intelligence Artificielle s'est imposée naturellement puisqu'il s'agit de formaliser une expertise , donc de construire une base de connaissances , avec ses règles , ses procédures, ses faits et sa structure déductive.

Le langage IA retenu est le Turbo-Prolog ,les raisons de ce choix sont présentées dans le mémoire.

Archer est un processeur d'aide à la modélisation,qui, à partir d'une description du système physique en langage prroche du langage naturel , construit le modèle Bond-Graph et détermine l'équation d'état associé sous forme d'expressions formelles.

Archer prépare des données qui peuvent servir directement dans des logiciels de simulation,tels que Acsl ou Basile, et apparaît donc comme un pré-processeur de tels progiciels.

Le mémoire se compose de quatre chapitres.

Le premier chapitre présente un rappel succint,mais complet de l'outil Bond-Graph , les variables mises en jeu, les composants,et les procédures nécessaires pour construire le modèle Bond-Graph.

Le deuxième chapitre situe Archer dans le contexte informatique des logiciels existants utilisant l'approche Bond-Graph et la spécificité d'Archer est présentée.

Quelques définitions générales sur l'approche IA sont également rappelées.

Dans le troisième chapitre,nous présentons de façon détaillée Archer,en insistant sur son architecture,les différentes procédures,avec mise en oeuvre sur des exemples.

Le quatrième chapitre aborde le problème du couplage entre modèles prédéfinis,stockés dans une base de données,ce qui donne plus de souplesse à l'utilisateur et permet une description de systèmes physiques complexes.

CHAPITRE I :

L'OUTIL , LES BOND - GRAPHS .

CHAPITRE I : L'OUTIL ,LES BOND-GRAPHS.

I) Introduction

Ce chapitre présente , de façon assez succinte,les principes de base de la méthodologie Bond-graph .

Après avoir défini les symboles graphiques associés au transfert de puissance, au codage des phénomènes physiques et les variables généralisées mises en jeu , nous détaillons les procédures classiques pour la construction du modèle Bond_Graph d'un système mécanique en dimension 1 et d'un système électrique,et pour l'affectation de la causalité.

La méthode retenue pour la mise en équation d'état est ensuite rappelée.

Des exemples trés simples apparaissent à chaque étape .

II) Représentation des transferts de puissance

Considérons les deux sous systèmes A et B représentés figure I-1

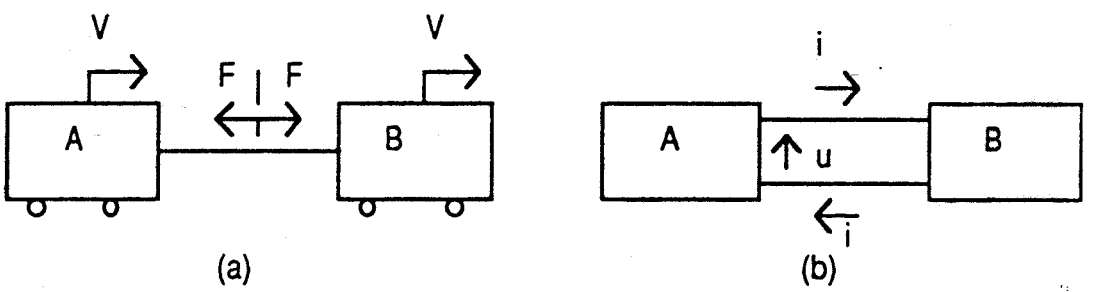


Figure I-1

Dans les deux cas (mécanique et électrique) , il existe une liaison physique entre A et B , soit par l'intermédiaire d'une barre (qu ' on supposera rigide et sans masse) dans le cas (a) , soit par un fil électrique (supposé sans perte) dans le cas (b) .

De plus , il y a transfert de puissance , mécanique pour (a) ou électrique pour (b) entre ces deux blocs .

Ces deux constatations vont être représentées par le symbole ——→ , qui correspond au "bond" (ou lien) du bond-graph .

La puissance instantanée échangée entre A et B se calcule , en mécanique , par $P = F V$ ou en électrique par $P = ui$.

Le lien porte les variables mises en jeu dans le calcul de puissance , et le sens de la demi -flèche est celui correspondant au produit P>0 .

Les schémas physiques de la figure 1 ont donc leur traduction Bond-Graph figure I-2 .

(a) (b)

Figure I-2

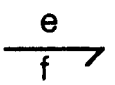
III) Variables mise en jeu

III-1) Variables de puissance

Nous avons vu précédemment que la puissance échangée P s'exprime par le produit de deux variables complémentaires u (ou F) et i (ou V) .

D'un point de vue général , indépendemment du domaine considéré , on parle de variables "généralisées" d'effort et de flux , notées respectivement e et f .

Ce sont les variables de puissance , et nous avons P = e f .

Par convention , nous représentons le lien toujours de la façon suivante  ou e  f .

La correspondance de ces variables de puissance pour différents domaines de la physique est présentée dans le tableau I-1 .

III-2) Variables d'énergie

L ' énergie est calculée par intégration de la puissance par rapport au temps

$$E(t)=\int P(t)\,d(t) \quad (E(0) \text{ supposée nulle })$$

On définit les variables d'énergie par les relations intégrales suivantes :

$$p(t)=\int_0^t e(\tau)\,d\tau \quad (p(0) \text{ supposée nulle })$$

$$q(t)=\int_0^t f(\tau)\,d\tau \quad (q(0) \text{ supposée nulle })$$

p(t) est le "moment généralisé" et q(t) est le "déplacement généralisé" .

Le tableau I-1 indique leur signification suivant les domaines de la physique .

Variables	Général	Electrique	Mécanique		Hydraulique
			Translation	Rotation	
Effort	e(t)	v tension	F force	τ couple	P pression
Flux	f(t)	i courant	V vitesse	ω vitesse angulaire	Q débit
Impulsion moment généralisé	$p=\int e\,dt$	λ flux	P impulsion	H moment angulaire	P_p intégrale de la pression
Déplacement	$q=\int f\,dt$	q charge	x déplacement	θ angle	V volume
Puissance	$P(t)=e(t)\,f(t)$	$v(t)\ i(t)$	$F(t)\ V(t)$	$\tau(t)\ \omega(t)$	$P(t)\ Q(t)$
Energie	$E(p)=\int f\,dp$	$\int i\,d\lambda$ magnétique	$\int v\,dP$ cinétique	$\int \omega\,dH$ cinétique	$\int Q\,dP_p$ cinétique
	$E(q)=\int e\,dq$	$\int e\,dq$ électrique	$\int F\,dx$ potentielle	$\int \tau\,d\theta$ potentielle	$\int P\,dV$ potentielle

Variables	Général	Electrique	Mécanique		Hydraulique
			Translation	Rotation	
Effort	e(t)	Volt (N-m)/C	N	N-m	N/m^2
Flux	f(t)	Ampère C/sec	m/sec	rad/sec	m^3/s
Impulsion moment généralisé	$p=\int e\,dt$	V-sec	N-sec	N-m-sec	$(N\text{-}sec)/m^2$
Déplacement	$p=\int f\,dt$	Coulomb A-sec	m	rad	m^3
Puissance	$P(t)=e(t)\,f(t)$	V-A Watt (N-m)/sec	(N-m)/sec	(N-m)/sec	(N-m)/sec
Energie	$E(p)=\int f\,dp$ $E(q)=\int e\,dq$	V-A-sec W-sec N-m	N-m	N-m	N-m

Tableau I-1

IV) Les éléments Bond-Graphs

Les éléments Bond-Graph simples peuvent se classifier de la façon suivante :

- éléments actifs : les sources
- éléments passifs : I , R , C
- éléments de jonction : 0 , 1 , TF , GY .

Nous ne présenterons pas ici les éléments multiports qui peuvent toujours être décomposés en éléments 1-ports simples (1 point de connection) .

IV-1) Eléments actifs : les sources

Ces éléments sont dits actifs car ils fournissent de la puissance au système .

On distingue
- les sources d'effort Se
- les sources de flux Sf

L'orientation de la demi-flèche est fixée et supposée toujours sortant de la source .

Se $\xrightarrow{e}$ ou Sf $\xrightarrow[f]{}$

Dans chaque cas , une des deux variables est supposée connue , et indépendante de la variable complémentaire induite qui dépend , elle , du système .

On a :

Se $\xrightarrow{e}$

Se : source d'effort e
avec e(t) donnée , cela peut être la gravité (mg)
ou une source de pression ...

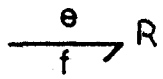
Sf $\xrightarrow[f]{}$

Sf : source de flux
avec f(t) donnée , cela peut être une source de débit
ou une source de courant ...

IV-2) Les éléments passifs : R , C , I

Ces éléments sont dits passifs car ils dissipent de la puissance (la demi-flèche sera toujours représentée entrant dans ces éléments) , "1-port" car ils ont un seul point de connection .

a) élément dissipatif d'énergie

* L'élément R : 

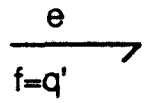
Cet élément est utilisé pour modéliser tout phénomène physique liant l'effort et le flux , par une relation du type

$\phi_R(e,f)=0$.

Cette loi peut être linéaire ou non linéaire ,et peut s'écrire en linéaire e = R f ou f = (1/R) e .

A titre d' exemple citons un amortisseur , une résistance électrique , ainsi que tout phénomène de frottement .

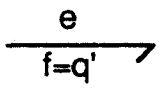
b) Les éléments de stockage d'énergie

* L'élément C : 

Cet élément est utilisé pour modéliser tout phénomène physique liant l'effort au déplacement :

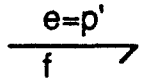
$\phi_C(e,q)=0$.

C'est un élément de stockage d'énergie (potentielle ou électrique) que l'on représente par



En linéaire la loi caractéristique s'écrit e = (1/C) q
ou q = C e . (Si on suppose les valeurs initiales nulles)

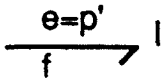
A titre d' exemple citons : ressort , accumulateur , condensateur , réservoir de stockage , tout phénomène d' élasticité ou de compressibilité .

* L'élément I : 

Cet élément est utilisé pour modéliser tout phénomène physique liant le flux au moment .

$\phi_I(f,p)=0$.

C'est un élément de stockage d'énergie(cinétique ou magnétique), noté :



La loi linéaire s'écrit :

$$f = (1/I) p \quad \text{ou} \quad p = I f$$

(Si on suppose les valeurs initiales nulles)

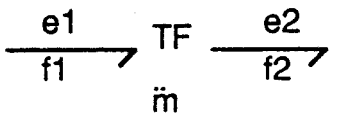
A titre d'exemple , citons les masses en translation , les inerties en rotation , les inductances ...

IV-3) Eléments de jonction

a) Les éléments transducteurs

Ce sont des éléments 2-ports , conservatifs de puissance .

* jonction TF



m est le module du transformateur .

Le transformateur est noté TF , et son module est défini par les relations suivantes :

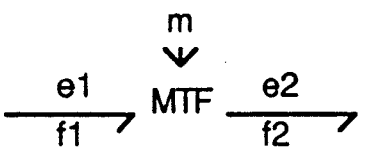
$$\begin{cases} e1 = m\, e2 \\ f2 = m\, f1 \end{cases}$$

C'est la loi générique qui caractérise le transformateur .

Comme exemple citons : le transformateur électrique , un système de poulies , un système d'engrenage, un bras de levier ...

Cette modélisation par un élément TF suppose bien sûr de poser à priori des hypothèses simplificatrices , en supposant négligeables des phénomènes d'inertie , de frottement , d'échauffement ... qui entraineraient des pertes de puissance .

Remarque : si m n'est pas constant , on parle de transformateur modulé



et on note MTF le transformateur modulé

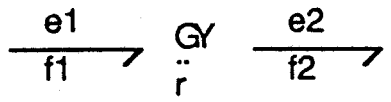
$$\begin{cases} e1 = m(t)\, e2 \\ f2 = m(t)\, f1 \end{cases}$$

Et comme exemples , citons :

- réducteur variable
- cinématique des mécanismes

* Soit la jonction GY

La représentation de l'élément "gyrateur"

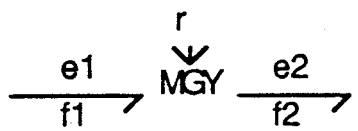


est associé à la loi

$$\begin{cases} e1 = r\, f2 \\ e2 = r\, f1 \end{cases}$$

où r est le module du gyrateur .

Si r est variable, on a un gyrateur modulé



associé à la loi générique suivante :

$$\begin{cases} e1 = r(t)\, f2 \\ e2 = r(t)\, f1 \end{cases}$$

Les composants physiques pouvant être modélisés pour un GY sont moins immédiats .

Il faut signaler que les éléments TF et GY sont trés importants pour représenter des transformations de puissance inter-domaines .

Ainsi on utilisera un élément TF pour représenter la transformation de la puissance hydraulique en puissance mécanique dans un verin hydraulique et le module du transformateur sera défini comme l'aire du piston du verin .

De même la transformation de la puissance électrique en puissance mécanique dans un moteur à courant continu se modélise par un GY de module égal au coefficient de couple du moteur .

b) Les éléments jonction-0 et jonction-1

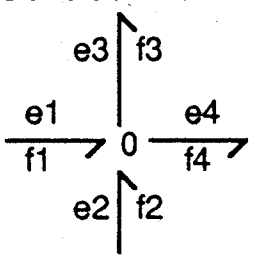
Ce sont des éléments conservatifs de puissance , multi-ports qui servent à interconnecter les éléments précédemment décrits .

* La jonction-0

Cette jonction sert à associer les éléments soumis au même effort .

La conservation de puissance fournit une relation entre les flux.

Considérons l'exemple suivant :



La loi générique , s'écrivant

égalité des efforts
somme pondérée des flux = 0

On a ici :

e1=e2=e3=e4
f1+f2-f3-f4=0

Les signes (+) ou (-) sur les fi dépendent de l'orientation , entrante ou sortante , des liens par rapport au 0 .

* La jonction-1

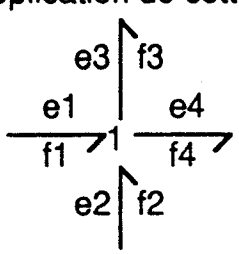
Cette jonction est la duale de la précédente .

Elle sert à associer les éléments soumis au même flux .

La loi générique est ici :

égalité des flux
somme pondérée des efforts = 0

L'application de cette loi à l'exemple suivant donne :



f1=f2=f3=f4
e1+e2-e3-e4 = 0

IV-4) Tableau récapitulatif

Tous ces éléments bond-graph , présentés rapidement ,sont regroupés dans le tableau I-2 , avec quelques exemples d'utilisation :

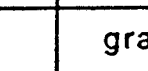
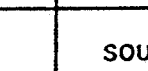
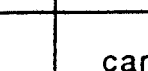
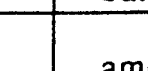
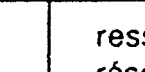
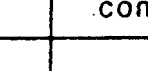
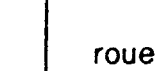
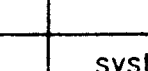
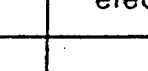
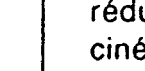
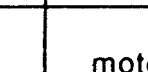
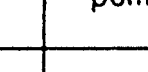
élément	nom	représentation	exemples
source	source e		gravité source de pression
	source f		source de débit source de courant
	source		caractéristiques d'une alimentation batterie, pompe
dissipation	résistance		amortisseur résistance électrique
stockage	capacité		ressort, accumulateur réservoire de stockage compressibilité
	inertie		masse inductance
transducteur	transformateur		roue et pignon vérin
	gyrateur		systémes électro-dynamiques
	transformateur modulé		réducteurs variables cinématique demécanisme
	gyrateur modulé		moteur électrique pompe centrifuge
jonction	jonction-0		force commune connection en parallèle
	jonction-1		effort commun connection en série

Tableau I-2

V) Procédure de construction d'un modèle Bond-Graph à partir d'un schéma physique .

La procédure de construction n'est pas la même suivant que le système étudié est mécanique , électrique ou hydraulique .

Nous présenterons ici le cas mécanique en dimension 1 et électrique , le domaine hydraulique étant tout à fait semblable au cas électrique .

(On parlera de noeuds de pression au lieu de noeud de tension)

V-1) Cas mécanique en dimension 1

Procédure :

1- Choisir un repère (axe x) qui servira pour l'orientation des demi-flèches .

2- Pour chaque vitesse que l'on souhaite utiliser , mettre une jonction-1 .

Y connecter les masses .

3- Placer des jonctions-0 entre les jonction-1 .

Ces jonctions serviront à représenter les relations entre vitesses .

Y placer les éléments R et C correspondants .

4- Placer les sources .

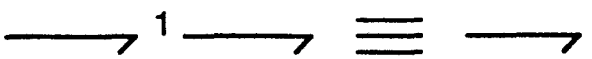
5- Relier les jonctions entre elles par des liens .

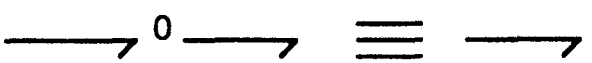
Vérifier l'orientation des demi-flèches en faisant le bilan des puissances (ou en vérifiant les relations entre vitesses , ce qui est équivalent) .

6- Simplifier le Bond-Graph si possible :

Les noeuds de vitesse nulle sont éliminés , ainsi que tous liens qui leur sont attachés .

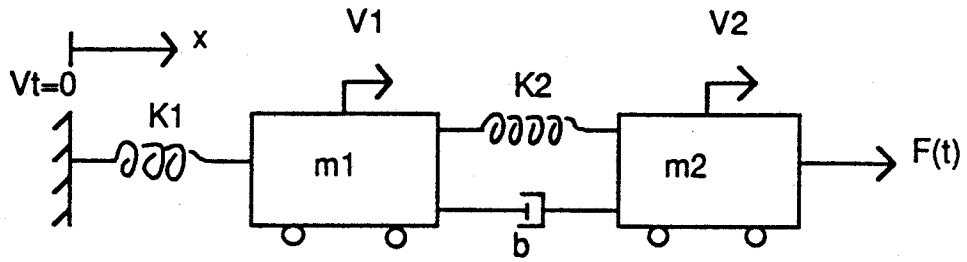
Deux liens attachés à une jonction , sans élément , peuvent être unifiés en un seul , dans les cas suivants :





Ne pas simplifier les situations suivantes :

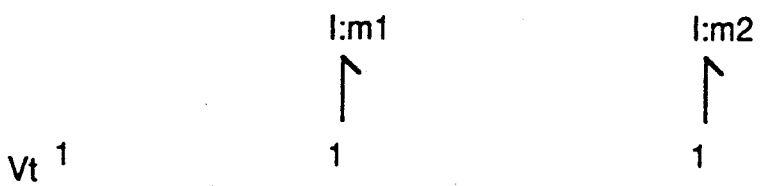
A titre d'exemple de mise en oeuvre de la procédure , considérons le système suivant



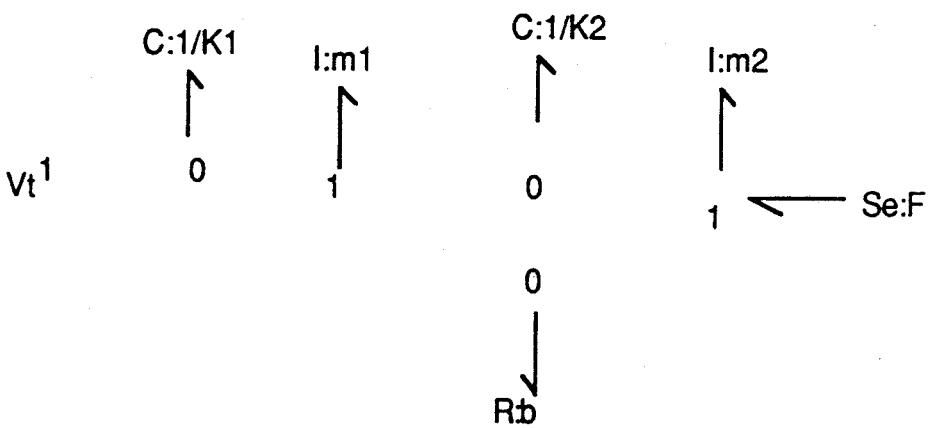
1 ère étape : choix de l'axe x
Hypothèse : V1>V2

2 ème étape : choix des vitesses caractéristiques
3 vitesses : V1 , V2 , Vt => 3 jonctions-1

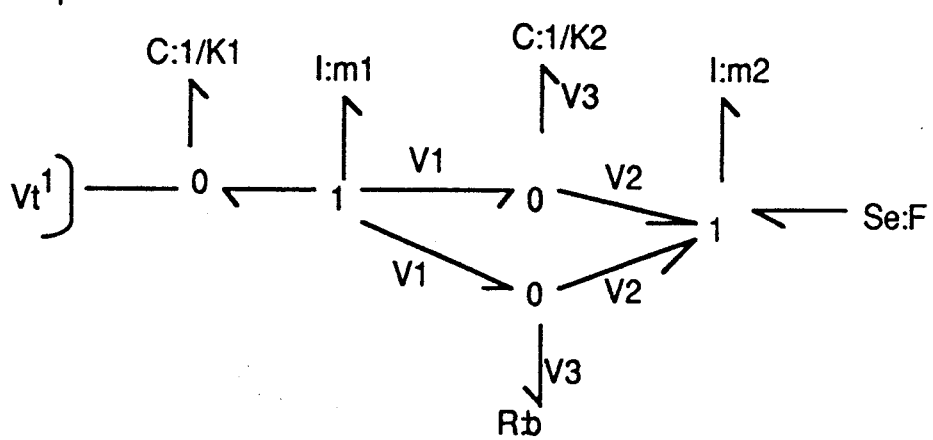
2 inerties : m1, m2 => connection des 2 masses



3 ème étape : placement des jonctions-0 entre les jonctions-1, puis placement des éléments R et C correspondants.
4 ème étape : placement des sources

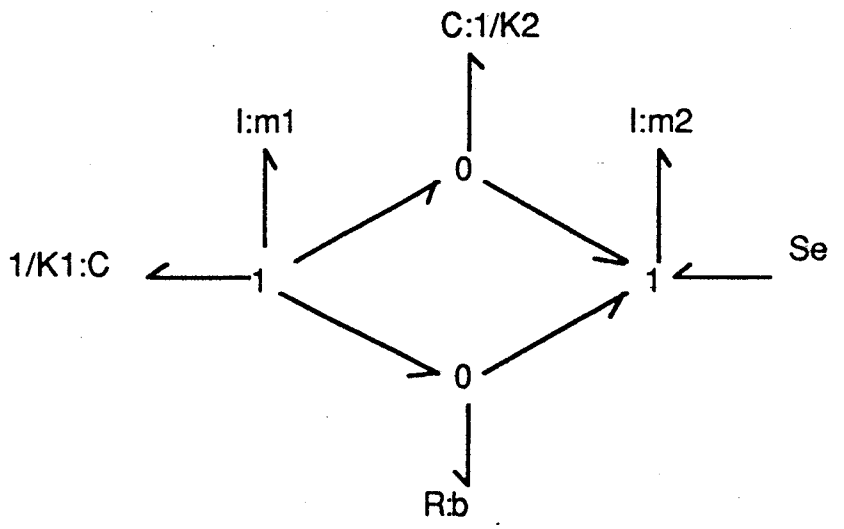


5 ème étape ,on relie les jonctions entre elles par des liens et l'on vérifie l'orientation des flèches en faisant le bilan des puissances .



Ici la vitesse d'élongation du ressort k2 et de l'amortisseur b est définie par V3 = V1 - V2 .

6 ème étape , la simplification du Bond-Graph si possible est réalisée.
On supprime le noeud 1 associé à la vitesse Vt nulle , ainsi que les liens qui y sont attachés .
On obtient ainsi le Bond-Graph final .



Il faut signaler que le Bond-graph de ce système n'est pas unique , et que l'on aurait pu obtenir le Bond-Graph suivant en choisissant de représenter sur un noeud unique 1 la vitesse commune V3 du ressort k2 et de l'amortisseur b , comme présenté Figure I-3 .

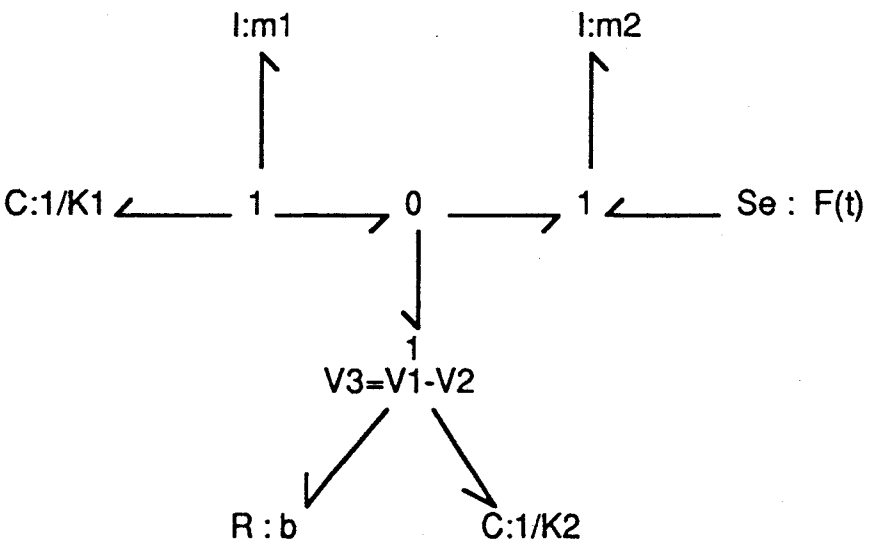


Figure I-3

V-2) Cas électrique

Procédure :

1- Fixer le sens de circulation du courant .

2- A chaque noeud du circuit de tension différente , mettre une jonction-0 .

3- Insérer sur des jonctions-1 placées entre les jonctions-0 , les éléments R,C,I .

4- Assigner le sens des flèches , qui correspond au sens du courant .

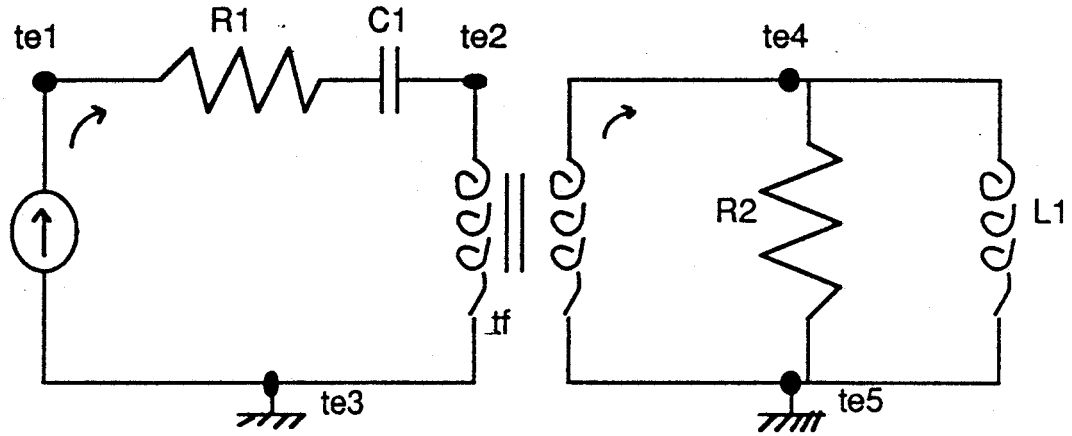
5- Choisir un point particulier comme masse .
Il sera le noeud de tension nulle .

6- Simplifier le Bond-Graph si possible .
- Les noeuds de tension nulle sont éliminés ainsi que tous les liens qui leur sont attachés .
- Deux liens attachés à une jonction sans élément , peuvent être unifiés en un seul ,dans les cas suivants :

Ne pas simplifier les situations suivantes

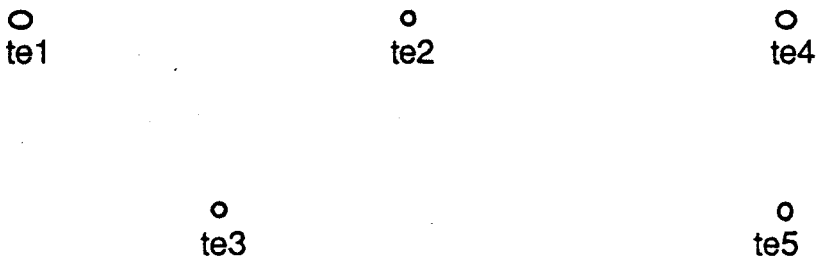
Remarque : il y a une certaine dualité dans les algorithmes entre les cas mécanique et électrique .

A titre d'exemple de mise en oeuvre de la procédure , considérons le système suivant .



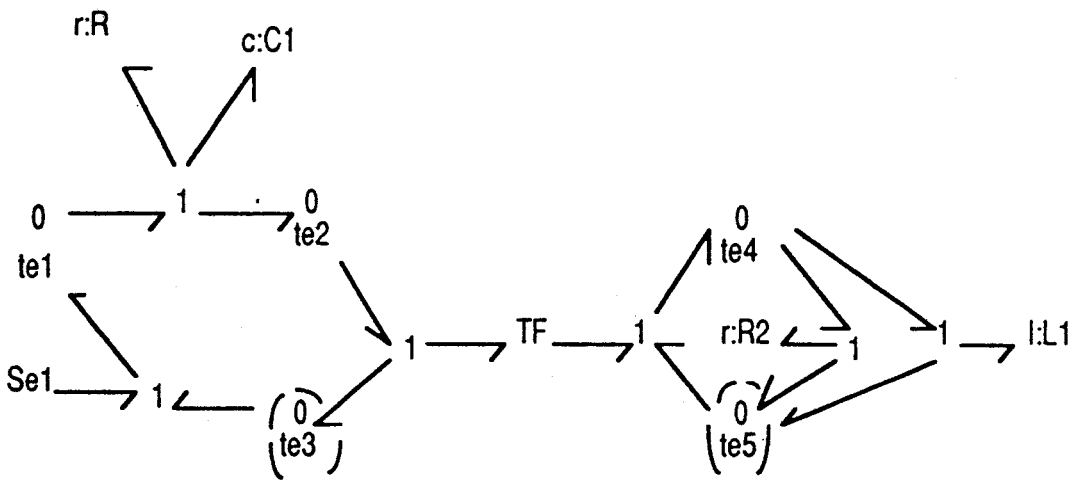
2 ème étape

Mise en place de jonction-0 à chaque noeud de tension



3 ème étape

On va insérer sur des jonctions-1 placée entre les jonctions-0 , les éléments R , C , I



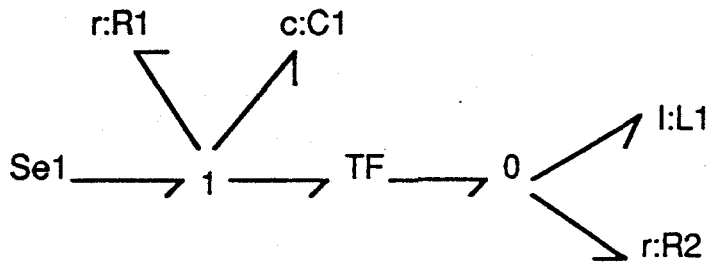
On opère aussi la 4 ème et la 5 ème étape

- affecter le sens des flèches en fonction du sens du courant .

- remarquer les points particuliers : points de masse .

6 ème étape : simplification du Bond-Graph si c'est possible .

On obtient :



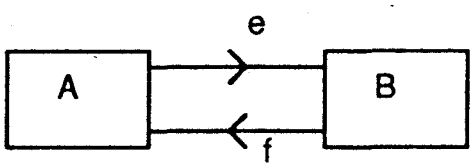
VI) Notion de causalité

Les Bond-Graphs correspondent à une structure topologique où sont représentés les échanges de puissance entre éléments .

On peut aussi définir une structure de calcul avec mise en évidence des relations de cause à effet au sein du système .

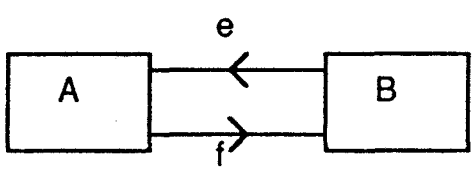
Lorsque deux sous-systèmes A et B sont couplés et échangent de la puissance instantanée P = e f , nous avons deux situations possibles :

- A applique à B un "effort" e , qui réagit en envoyant à A un "flux" f, soit



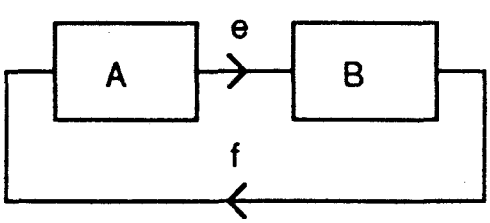
(a) Figure

- A envoie à B un "flux" f , qui répond par un "effort" e

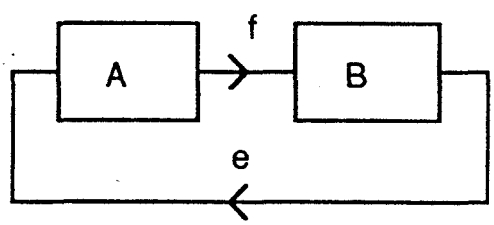


(b) Figure

Ces deux cas conduisent à deux schémas-bloc différents



(a)

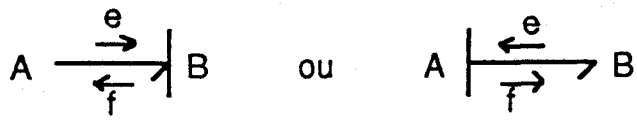


(b)

Pour prendre en compte ces relations de cause à effet et les représenter sur le modèle Bond-Graph ,nous introduisons le trait causal .

Le trait causal est placé verticalement au lien ,il indique par convention , le sens d'application de l'effort , le flux étant toujours considéré comme entrant dans le sens opposé au trait causal .

Ainsi si on reprend l'exemple précédent ,le transfert de puissance de A vers B conduit à deux positions du trait causal .

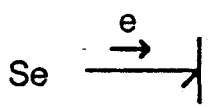


(a) Figures (b)

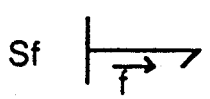
La position du trait causal est tout à fait indépendante du sens de la demi-flèche .

Elle n'est pas arbitraire mais soumise à des règles que nous présentons dans ce qui suit .

VI-1) Les causalités obligatoires



: source d'effort



: source de flux

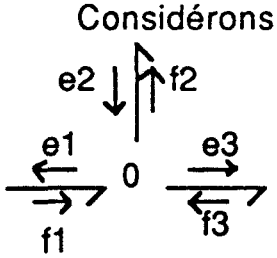
L'effort , respectivement le flux , sont supposés toujours connus pour le système , et sont donc des données .

VI-2) Restrictions de causalité

* Jonctions 0 et 1

. Jonction-0 (effort commun)

Considérons l'exemple suivant



La loi générique , correspondant au bilan des puissances s'écrit :

$$\begin{cases} e1=e2=e3 \\ f1-f2-f3=0 \end{cases}$$

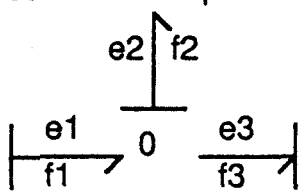
Cette écriture ne s'adapte pas directement au calcul .

Il faut d'une part définir quel est l'effort qui donne sa valeur aux autres et d'autre part définir quel est le flux à calculer et quels sont les flux de valeurs connues .

Supposons que e2 est connu ; nous pouvons écrire

e1 = e2

e3 = e2 ce qui conduit à positionner les traits causaux comme indiqué sur la figure

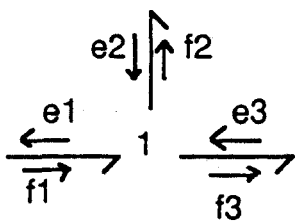


Ceci donne directement l'information sur les flux , ce qui permet d'écrire f2 = f1 - f3

A une jonction 0, un seul effort peut donner sa valeur aux autres , ce qui conduit à la règle suivante :"1 seul trait causal prés de la jonction 0" .

. Jonction-1 (flux commun)

Considérons l'exemple suivant présenté figure :



La loi générique , associé au sens des demi-flèches est

$$\begin{cases} f1=f2=f3 \\ e1-e2-e3=0 \end{cases}$$

(bilan des puissances)

Le raisonnement est le même que pour la jonction-0 .

Un seul flux donne sa valeur aux autres .

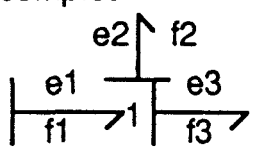
Supposons que ce soit f1 , nous pouvons écrire

f2 = f1

f3 = f1

ce qui donne le Bond-Graph complet

et conduit à écrire



e1=e2+e3

La règle d'affectation de la causalité est ici : " 1 seul lien sans trait causal près du 1 " .

* Jonction TF et GY

. jonction TF

Nous avons deux possibilités d'affectation de la causalité :

Les lois génériques respectives sont :

$f2 = m\ f1$
$e1 = m\ e2$ avec f1 et e2 connus

$f1 = (1/m)\ f2$
$e2 = (1/m)\ e1$ avec f2 et e1 connus

Nous remarquons qu'il y a un trait causal prés du transformateur .

. jonction GY

Nous avons deux possibilités d'affectation de la causalité

Les lois génériques respectives sont :

$e1 = r\ f2$
$e2 = r\ f1$ avec f1 et f2 connus

$f2 = (1/r)\ e1$
$f1 = (1/r)\ e2$ avec e1 et e2 connus

Nous remarquons qu'il y a soit deux traits causaux ,sur le gyrateur , soit pas de trait causal dessus .

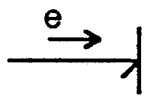
VI-3) Causalités préférentielles

Elles concernent les éléments C et I mettant en jeu des relations de type intégrale ou dérivée .

. élément C

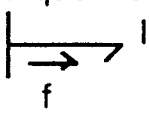
Ecrire $e = (1/C) \int f\,dt$ suppose que f est connu pour l'élément C , ce qui se représente par 

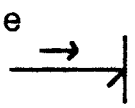
.

Ecrire $f = c\,(de/dt)$ suppose que e est connu pour l'élément C , ce qui se représente par  C .

Pour des considérations d'ordre numérique (il est plus aisé et plus robuste d'intégrer que de dériver) , on essaiera d'affecter à l'élément C une causalité dite "intégrale" associée à une loi de type intégrale .

. élément I

Ecrire $e = I\,(df/dt)$ suppose que f est connu pour l'élément I ,ce qui se représente par  I .

Ecrire $f = (1/I) \int e\,dt$ suppose que e est connu pour l'élément I , ce qui se représente par  I .

Pour les mêmes raisons que pour l'élément C , on essaiera d'affecter à l'élément I une causalité dite "intégrale" associée à une loi de type intégrale .

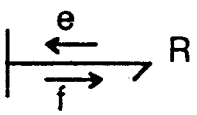
Remarque :

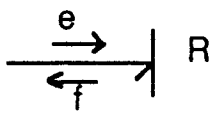
Dans certains cas , il est impossible d'affecter à tous les I et C une causalité intégrale .

L'apparition de causalités dérivées conduit toujours l'utilisateur à s'interroger , et peut dans certains cas s'interpréter comme une description trop grossière des phénomènes mis en jeu .

. élément R

Dans le cas linéaire , deux situations tout à fait équivalentes peuvent se présenter .





$e = R\,f$	$f = (1/R)\,e$
f connu	e connu
"causalité flux"	"causalité effort"

Dans ce cas , pour l'élément R ,il n'y a pas de causalité préférentielle , il s'adapte selon la situation restrictive du contexte .

Lorsque l'élément R est non linéaire (restriction hydraulique par exemple) , la causalité n'est plus arbitraire et peut apparaitre comme une contrainte obligatoire comme pour les sources .

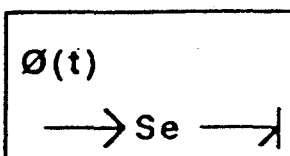
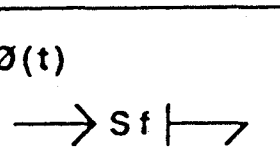
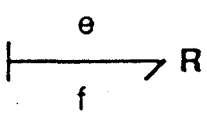
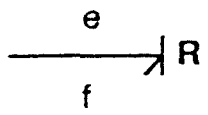
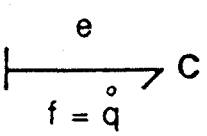
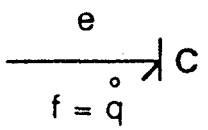
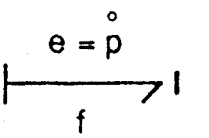
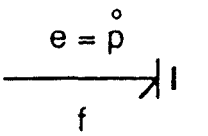
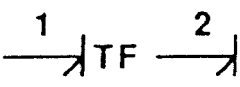
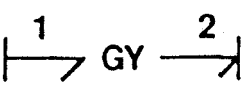
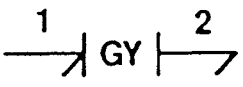
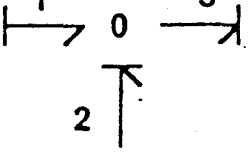
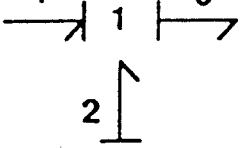
	$e = \emptyset(t)$		$f = \emptyset(t)$
	$e = R * f$		$f = \frac{1}{R} e$
	$e = e(0) + \frac{1}{C}\int f\, dt$		$f = C\frac{de}{dt}$
	$e = I\frac{df}{dt}$		$f = f(0) + \frac{1}{I}\int e\, dt$
	$e1 = m * e2$ $f2 = m * f1$		$e2 = \frac{1}{m} e1$ $f1 = \frac{1}{m} f2$
	$e1 = r * f2$ $e2 = r * f1$		$f1 = \frac{1}{r} e2$ $f2 = \frac{1}{r} e1$
	$e1 = e2$ $e3 = e2$ $f2 = -f1 + f3$		$f1 = f2$ $f3 = f2$ $e2 = e3 - e1$

Tableau I-3

VI-4) Procédure d'affectation de la causalité

1 - affecter la causalité aux sources .

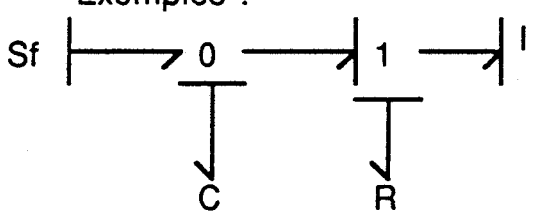
2 - Mettre tous les I et tous les C en causalité intégrale .

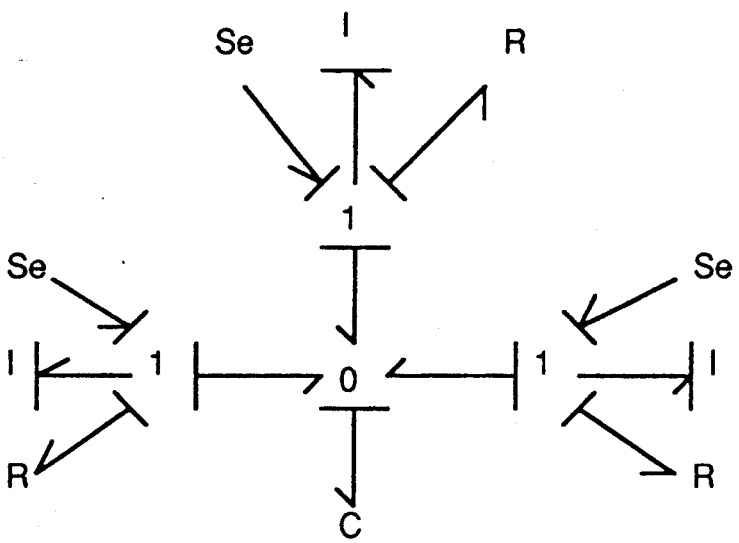
3 - Affecter les causalités aux jonctions 0 et 1 .

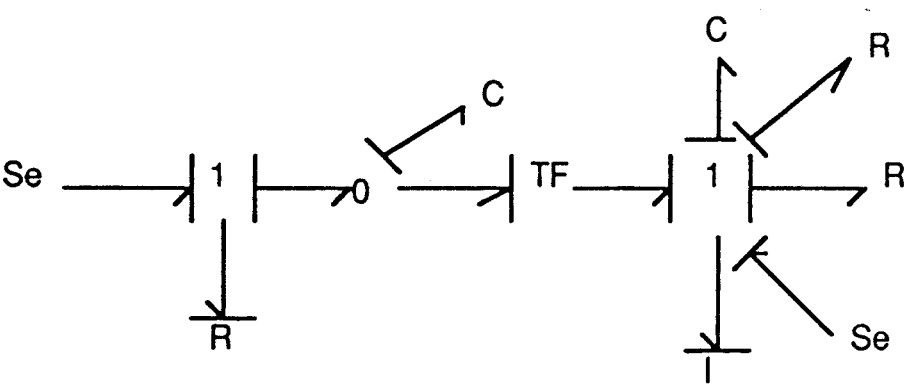
4 - Affecter les causalités aux éléments R , TF , GY en fonction des possibilités restantes .

5 - Voir les conflits de causalité .
Cas de conflit , reprendre en 2 et modifier la causalité sur l'élément origine du conflit .

Exemples :

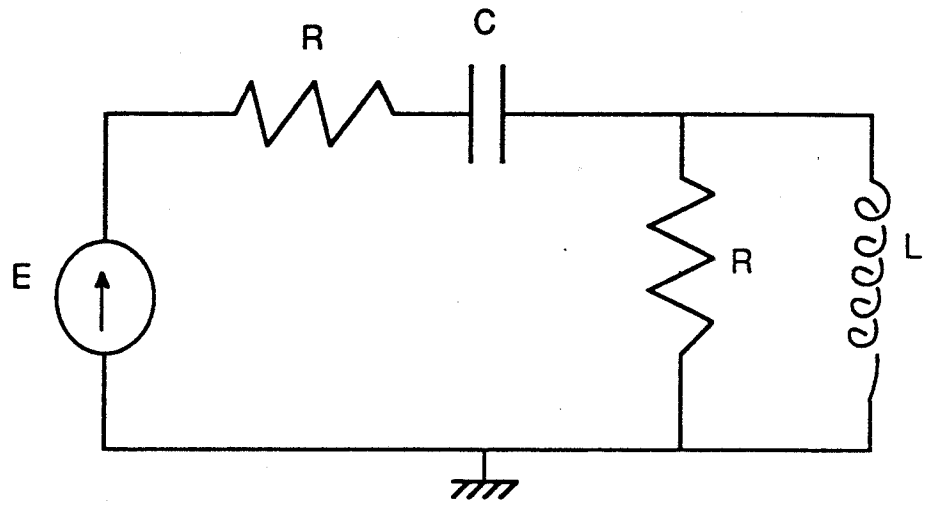


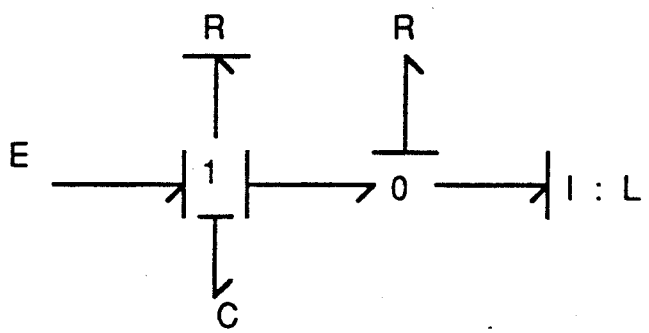


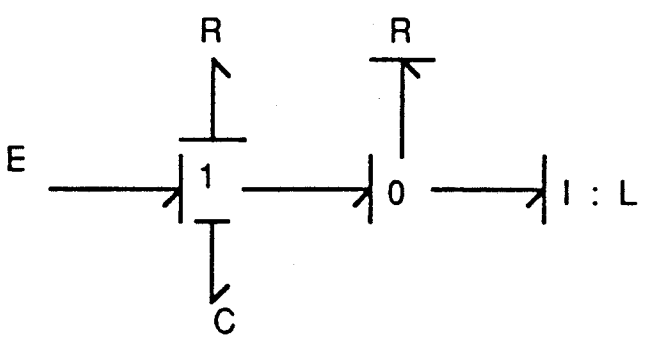


2 cas particuliers :

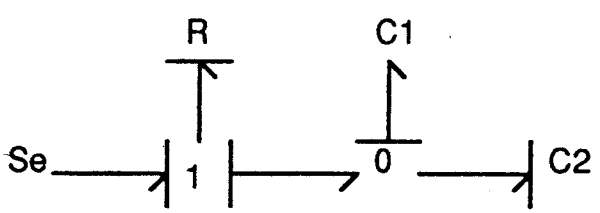
1 er cas : 2 affectations de causalité possibles .







2 ème cas : causalité dérivée



VII) Détermination de l'équation d'état .

VII-1) Variables d'état .

. Elles sont des variables douées de mémoire (liée à la phase d' intégration)

. Elles sont des variables d'énergie

--> Les variables d'état sont associées aux éléments I et C , et sont regroupées en $x=\begin{pmatrix} P \\ q \end{pmatrix}$

Seules leurs dérivées ,

$\dot{x}=\begin{pmatrix} \dot{p}=e \text{ sur } I \\ \dot{q}=f \text{ sur } C \end{pmatrix}$

apparaissent sur le Bond-Graph .

Elles sont directement attachées aux composants physiques.

Si tous les I et C sont en causalité intégrale alors la dimension de X est n , égale au nombre de I et C .

Si le Bond-Graph contient n2 éléments en causalité dérivée , alors le vecteur état X devient

$X=\begin{pmatrix} x_i \\ x_d \end{pmatrix}$ avec $x_i \in R^{n1}$ $x_d \in R^{n2}$ $n1+n2=n$

et le système est d'ordre n1 .

VII-2) Méthode générale de mise en équation

Un Bond-Graph peut se représenter d'une façon générale en regroupant les éléments en champs de vecteurs , comme présenté figure I-4.

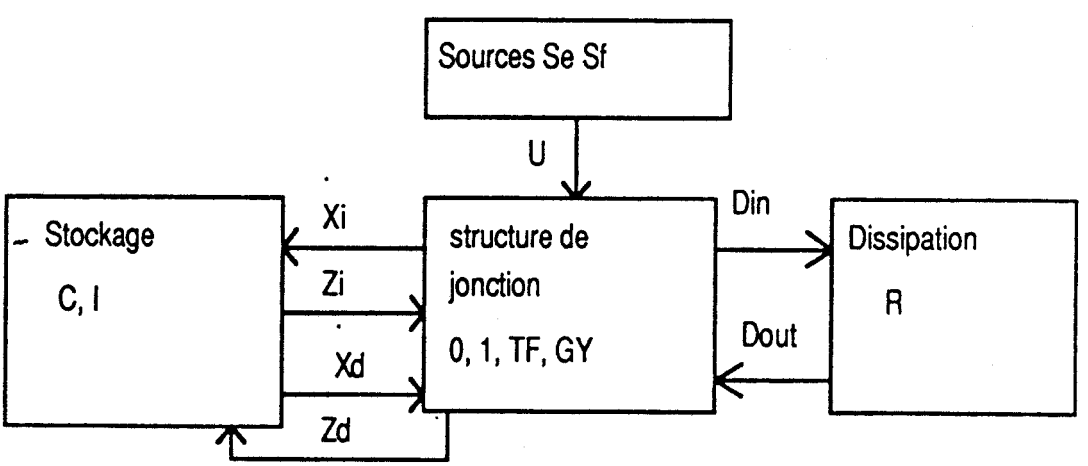


Figure I-4

Les vecteurs mis en oeuvre sont :

$\dot{x}_i = \begin{pmatrix} e \text{ sur } I \\ f \text{ sur } c \end{pmatrix}$ Dérivée du vecteur état associé aux éléments I et C en causalité intégrale

$\dot{x}_d = \begin{pmatrix} e \text{ sur } I \\ f \text{ sur } C \end{pmatrix}$ Dérivée du vecteur état associé aux éléments I et C en causalité dérivée

$\rightarrow Z_i = \begin{pmatrix} f \text{ sur } I \\ e \text{ sur } C \end{pmatrix}$ Vecteur état complémentaire associé à x_i

$\rightarrow Z_d = \begin{pmatrix} f \text{ sur } I \\ e \text{ sur } C \end{pmatrix}$ Vecteur état complémentaire associé à x_d

$U = \begin{pmatrix} Se \\ Sf \end{pmatrix}$

Din = éléments e et f entrant dans le champ dissipatif R
Dout = éléments e et f venant du champ R

. Les lois élémentaires associées aux éléments I,C,R permettent d'écrire :
Zi = Fi Xi (cas linéaire)
avec Fi définie positive et diagonale composée de termes 1/I et 1/C en causalité intégrale
ou Zi = Fi (Xi) (cas non linéaire)

Dout = L Din
avec L définie diagonale , composée de 1/R et R
ou Dout = L (Din)

Xd = Fd Zd
avec Fd définie diagonale
composée de termes I et C associés aux éléments en causalité dérivée
ou Xd = Fd (Zd)

Les relations liées à la structure du système s'écrivent sous forme matricielle , en considérant les variables entrant dans la structure de jonction $(Z_i, \dot{x}_d, Dout, U)$ et les variables sortant de la structure de jonction $(\dot{x}_i, Din, Z_d)$.

Soit :

$$\begin{pmatrix} \dot{x}_i \\ Din \end{pmatrix} = \begin{pmatrix} S11 & S12 & S13 & S14 \\ S21 & S22 & S23 & S24 \end{pmatrix} \begin{pmatrix} Z_i \\ \dot{x}_d \\ Dout \\ U \end{pmatrix}$$

S

S est appelée matrice de la structure de jonction et est composée exclusivement de 0,+1 , -1 , m(TF) , r(GY) .

Cette matrice est constante dans le cas linéaire , mais peut comporter des termes variables si m et r non constants .

Le vecteur état complémentaire Zd , associé aux éléments I et C en causalité dérivée s'exprime en fonction de Zi et U par

$$Zd = E1\ Zi + E2\ U$$

avec $E1 = -(S12)^T$

En combinant les différentes expressions,on obtient,dans le cas linéaire,(en supposant toutes les matrices constantes)

$$w\dot{x}_i = Ax_i + B\ U + E\ \dot{U}$$

avec

$$A = \left(S11 + S13\ L\ (I - S23\ L)^{-1} S21\right) F_i$$

$$B = \left(S14 + S13\ L\ (I - S23\ L)^{-1} S24\right)$$

$$w = I + \left(S12 + S13\ L\ (I - S23\ L)^{-1} S22\right) F_d^{-1}\ S12^T\ F_i$$

$$E = \left(S12 + S13\ L\ (I - S23\ L)^{-1}\ S22\right) F_d^{-1}\ E2$$

Avec le changement de variable défini par

$W\ Xi = W\ Ti + E\ U$

la nouvelle équation d'état devient

$$\dot{T}_i = A^*\ T_i + B^*\ U$$

alors

$$A^* = w^{-1}\ A$$

$$B^* = w^{-1}\left(A\ w^{-1}\ E + B\right)$$

le vecteur de sortie y est:

$$y=(S31\ S32\ S33\ S34)\begin{pmatrix} z_i \\ \dot{x}_d \\ Dout \\ U \end{pmatrix}$$

Cette équation d'état se simplifie dans les cas suivants:

-s'il n'y a pas d'éléments I ou C en causalité dérivée, alors W=identité

E=0

-Si les éléments en causalité dérivée ne sont pas liés aux sources , alors E2=0 et E=O

-Si les éléments en causalité dérivée ne sont pas liés aux éléments R , alors S22=0 .

VIII) Conclusion

Tous les principes de base de l'approche Bond-Graph rappelés dans ce chapitre constituent l'expertise du Bond-Graphiste et seront utilisés par la suite lors de la présentation d'Archer .

CHAPITRE II
LE CONTEXTE INFORMATIQUE .

Chapitre II Le contexte informatique

I) Introduction

De nombreux logiciels existent actuellement sur le marché, ou sont en cours de développement,qui utilisent l'approche Bond-Graph.

Dans une première partie ,nous présentons rapidement les caractéristiques des logiciels les plus connus,et nous insistons sur la position d'Archer dans ce contexte, en présentant les points qui lui sont spécifiques.

Dans une deuxième partie,nous rappelons les définitions et les composants de base d'un système expert.

Ensuite nous présentons l'organisation générale d'Archer.

II) Logiciels utilisant l'approche Bond-Graph

Tous les logiciels existant actuellement sur le marché utilisent le modèle Bond-Graph comme modèle de départ.

L'utilisateur construit le Bond-Graph du système étudié, et l'introduit,avec un codage lié au logiciel,pour obtenir des courbes de comportement du modèle soumis à des entrées fixées.

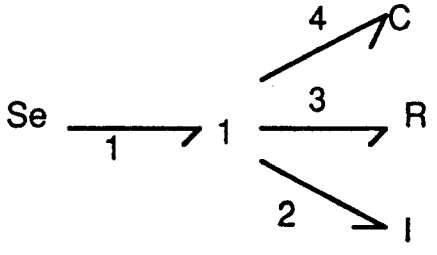
Ces logiciels sont donc essentiellement des logiciel de simulation.

Ils sont développés en langage structuré , essentiellement le FORTRAN,et aucun de ceux que nous connaissons n'utilise la démarche IA .

II-1 ENPORT [Rosenberg 1974]

Ce logiciel est le plus ancien , et fut développé pour de gros systèmes informatiques par Rosenberg , un des pères fondateurs des Bond-Graphs.

Le Bond-Graph est entré sous forme de code , tel que
Se1 , I2 , R3 , C4 , 1 1234
pour le Bond-Graph:



ENPORT affecte les causalités , et construit l'équation d'état du système à partir des valeurs numériques des éléments.

Il réalise ensuite l'intégration de l'équation et fournit la réponse du système , pour une entrée donnée , sous forme de table de résultats ou de courbe de réponse .

II-2 TUTSIM

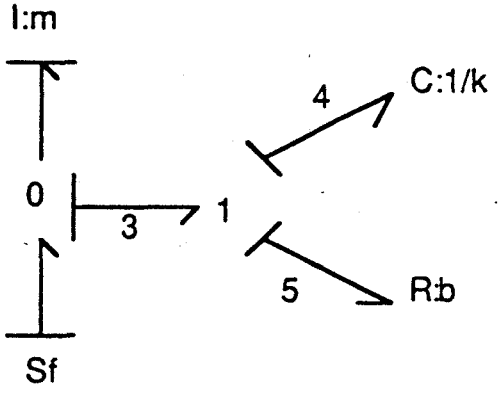
Ce logiciel a été développé à l'université de Enschede , dans l'équipe du professeur Van Dixhoorn [Beukeboom 1985].

Il est implanté sur petits calculateurs , de type PC ou sur Vax.

Il accepte en entrée des bloc-diagrammes (linéaires ou non),des Bond-Graphs ou une combinaison des deux et calcule les réponses du modèle à des entrées données .

C'est un programme trés convivial,encore trés proche des principes de base de la simulation analogique , ce qui en fait un excellent outil pédagogique.

L'utilisateur construit le schéma de simulation du système à partir des équations différentielles issues de l'application des lois de la physique , ou le modèle Bond-Graph correspondant complet (avec causalités) et introduit les données sous forme de code , comme présenté sur la figure II-1.



(a)

```
1 , Sf
2 , I , 4 , 5
4 , C , 1 , -2
5 , R , 1 , -2
```

(b)

Figure II-1 Modèle Bond-Graph et son code pour Tutsim

II-3 CAMP [Granda 1985]

CAMP est développé pour traduire la représentation Bond-Graph d'un système physique en codes Fortran utilisables par des logiciels de simulation comme IBM DSL ,CSMP,ACSL ou CSSL.

Il génère à partir du modèle Bond-Graph , les équations différentielles du premier ordre (correspondant à l'équation d'état) , les variables d'état , et crée les fichiers appropriés pour la simulation.

Le modèle Bond-Graph est entré sous CAMP grâce à un code trés proche de celui de ENPORT , c'est à dire , une liste des éléments et du numéro du lien auquel ils sont attachés .

L'affectation des causalités se fait automatiquement

II-4) SCRIBT

Ce logiciel , développé par le CEA et la Cisi est un préprocesseur de NEPTUNIX et necessite des moyens informatiques importants.

Il est dédié à la robotique essentiellement .

L'utilisateur assemble des icônes correspondant à des éléments Bond-Graph et Scribt transforme ces modèles en langage prêt pour la simulation , sous forme d'équations dynamiques.

II-5) SIDOPS [Broenink Jf 1986]

SIDOPS est le processeur de description de modèles de CAMAS,qui accepte en entrée à la fois des modèles Bond-Graph et des relations constitutives.

Il est considéré par son auteur comme un langage de construction de modèles Bond-Graphs.

SIDOPS affecte la causalité au Bond-Graph et construit le modèle mathématique associé , sous forme d'équations différentielles explicites $\dot{x}_k = f(x_k , u_k)$ ou implicite $\dot{x}_k = f(x_k , \dot{x}_k , u_k)$ suivant que le modèle possède des causalités dérivées.

Ces équations sont ensuite traitées et intégrées par CAMAS.

II-6) SIMODO [Bresch 1986]

Le but de ce logiciel est de traduire un Bond-Graph en équation d'état.

Il se compose d'un éditeur graphique d'entrée des modèles Bond-Graph,d'un module d'analyse et d'assemblage de modèles et d'un modèle de mise en forme des modèles mathématiques.

Ce travail a fait l'objet d'un mémoire de thèse en 1986,et nous n'avons pas trouvé de suite depuis .

II-7) Spécificité d'ARCHER

ARCHER est un processeur d'aide à la modélisation.

Il permet de construire des modèles mathématiques de systèmes physiques , et en ce sens on peut le comparer à CAMP, SCRIBT ou SIDOPS .

Il ne réalise pas la simulation de l'évolution des modèles et peut donc être considéré plutôt comme un préprocesseur pour des logiciels comme ACSL (Rapid Data) ou Basile (Inria -Simulog) .

Les principales caractéristiques qui le distinguent de ces logiciels sont les suivantes:

*ARCHER ne demande pas à l'utilisateur de connaître l'outil Bond-Graph.

La description du système se fait à partir du schéma physique, grâce à un langage utilisateur proche du langage naturel.

Lorsque le système est complexe, et ne peut pas être décrit simplement par des composants élémentaires, nous envisageons de construire des bases de données de modèles prédéfinis, spécifiques du domaine physique concerné, qui seront couplés entre eux , de manière tout à fait transparente pour l'utilisateur.

Bien sûr,un code Bond-Graph est disponible pou Bond-Graphistes qui voudraient utiliser directement les modèles des bases de données.

* ARCHER est le seul processeur utilisant l'approche Intelligence Artificielle,ce qui lui donne une grande souplesse.

Ainsi les modèles mathématiques obtenus se présentent sous forme d'expressions formelles,les paramètres intervenant dans la modélisation ne sont à définir que lors de la phase de simulation.

*ARCHER dessine automatiquement à l'écran le modèle Bond-Graph,en respectant des critères d'esthétique et de sens physique [Bouayad 1990] , ce qui est unique,les autres logiciels ne permettant que le dessin à l'écran par l'utilisateur à partir d'icones prédéfinies.

*ARCHER est aussi un outil d'analyse qui tire de la richesse de l'outil Bond-Graph toutes les informations concernant les propriétés structurelles du modèle (commandabilité , observabilité,stabilité asymptotique).

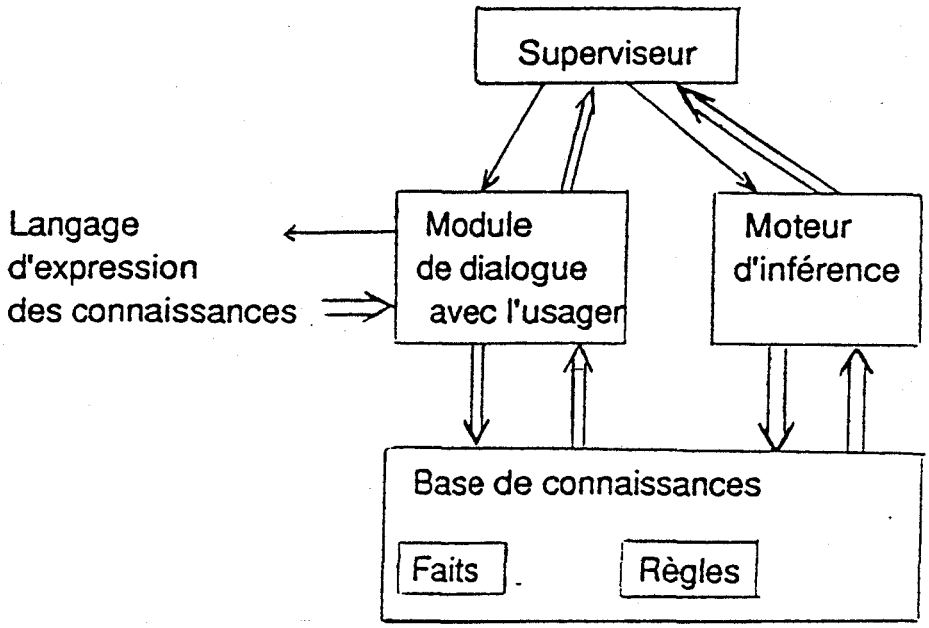
Les modèles mathématiques sont construits directement à partir de l'analyse causale du Bond-Graph effectuée sur les faits obtenus dans la base de connaissance [Azmani 1990].

III) Intelligence artificielle , système expert

III-1)Présentation générale d'un système expert .

Un système expert est un programme conçu pour raisonner habilement à propos de tâches dont on pense qu'elles requièrent une expertise humaine considérable .

L'organisation de principe d'un système expert se présente sous l'approche donnée par le schéma II-1.



=> : flèches doubles -> données et résultats
-> : flèches simples -> commande

Schéma II-1

III-2) Fonctionnement d'un système expert.

Schématiquement , considérons le principe de fonctionnement d'un système expert:

- Une base de connaissances:

Ces connaissances correspondent à des règles de raisonnement qui ont été tirées d'un certain domaine de connaissances:diagnostic médical,analyse de circuit,exploration minière,etc...

- Une base de faits

Ce sont des faits vérifiés dont on prend connaissance.
Exemple : des résultats d'analyses pour un patient donné.

- Un but (question posée)

C'est un fait dont on veut savoir s'il peut être déduit à partir de la base de faits en appliquant des règles de la base de connaissances.

- Un moteur d'inférences ou "machine déductive"

C'est le mécanisme qui va tenter de résoudre le problème posé .

Sommairement , on peut considérer deux types de moteurs d'inférences, ceux qui procèdent par chainage avant et ceux qui procèdent par chainage arrière.

-->Chainage avant:

On part de la base de faits et on déclenche des règles dont les prémisses sont entièrement contenus dans cette base de faits.

On obtient ainsi une nouvelle base de faits et on poursuit jusqu'à ce que soit on tombe sur le but ,soit plus aucune règle ne puisse s'appliquer.

Bref,on essaie d'agrandir la base de faits en appliquant successivement des règles de la base de connaissance,de façon à ce que la base de faits finisse par englober le fait but, si c'est possible.

Le principe de chainage avant repose sur ce que l'on appelle en logique le modus ponens : si (A->B) et A ,alors B.

-->Chainage arrière

On regarde les règles qui ont le but fixé dans leur conséquences.

On considère chacune de ces règles,si l'une d'elles a toutes ses prémmisses dans la base de faits,on a succés , sinon on considère les prémisses comme de nouveaux buts et on recommence.

Le principe du chainage arrière repose sur ce que l'on appelle le modus tollens:
si (A->B) et non B alors non A.

III-3) Organisation d'un système expert

L'organisation fonctionnelle d'un système expert est représentée sur le schéma suivant,où apparait la structure modulaire associée .

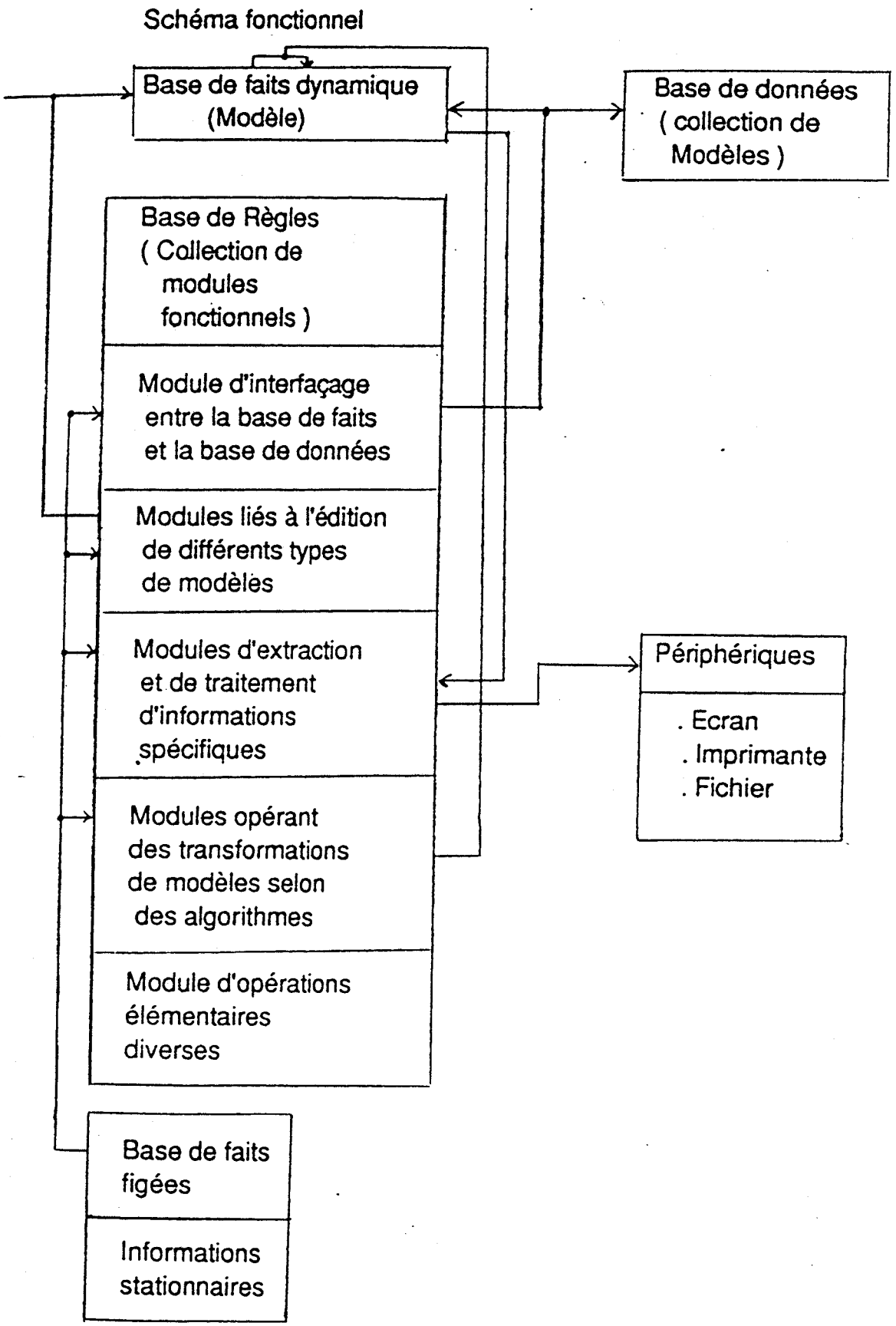


Schéma fonctionnel
Base de faits dynamique (Modèle)
Base de données (collection de Modèles)
Base de Règles (Collection de modules fonctionnels)
Module d'interfaçage entre la base de faits et la base de données
Modules liés à l'édition de différents types de modèles
Modules d'extraction et de traitement d'informations spécifiques
Modules opérant des transformations de modèles selon des algorithmes
Module d'opérations élémentaires diverses
Périphériques
. Ecran
. Imprimante
. Fichier
Base de faits figées
Informations stationnaires

Schéma II -2

IV) Présentation générale d'Archer

IV-1) Historique

Une première maquette d'Archer, représentant une étude de faisabilité,a été réalisée sous Vax 750 en Prolog .

Une structure arborescente , en mondes et sous-mondes donnait au projet une structure hiérarchisé comme indiquée schéma II-3 .

Soit l'arborescence explorable suivante :

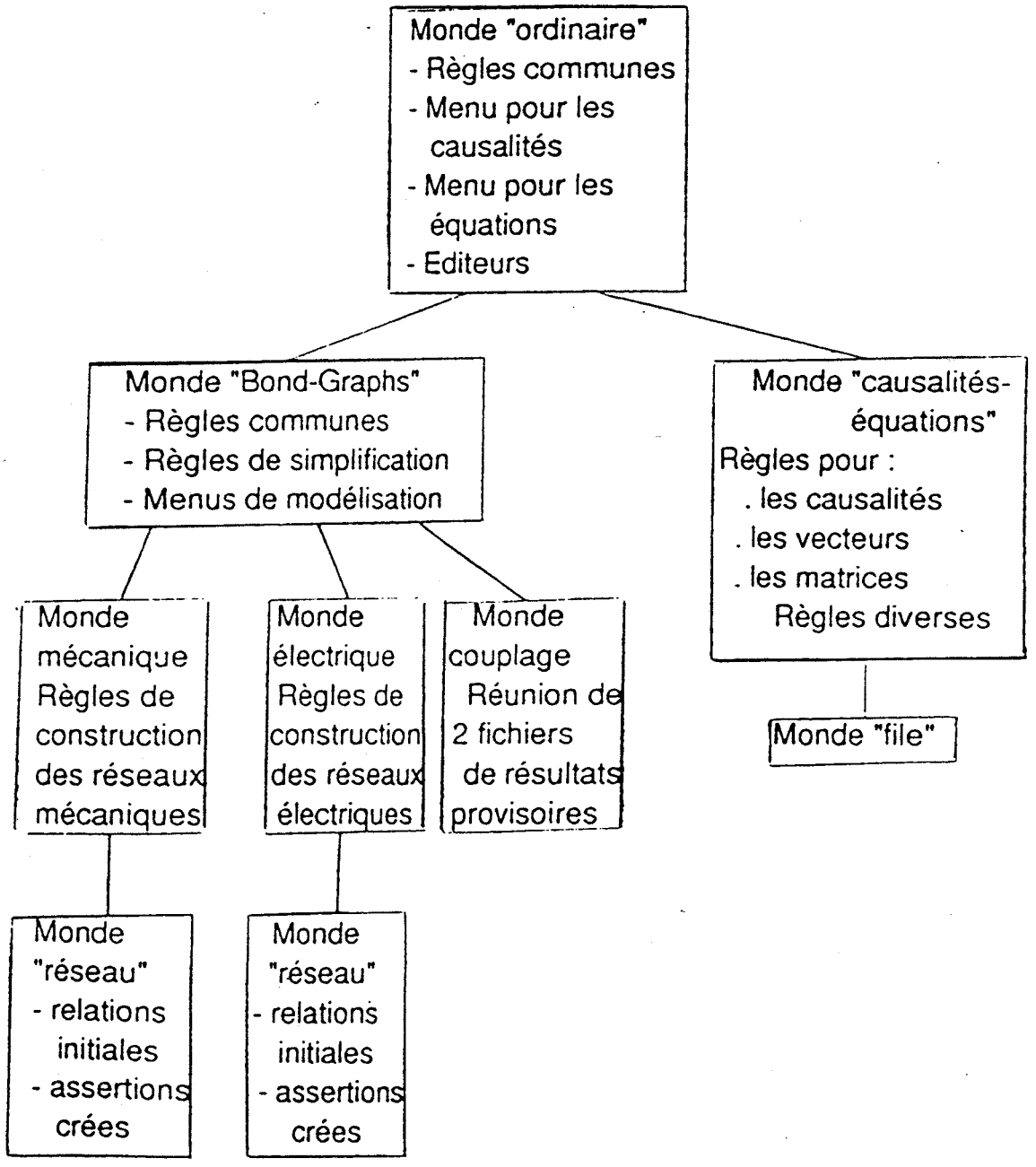


Schéma II-3

Pour des raisons de portabilité de souplesse d'utilisation , et à cause des performances graphiques que nous recherchions , nous avons choisi de réécrire Archer en Turbo Prolog et de l'implanter sur IBM-PC AT.

Deux versions successives,liées aux différentes versions de Turbo Prolog , ont existé .

Nous présentons ici la version actuelle, écrite en Turbo Prolog 2 qui se présente sous forme trés modulaire.

IV-2) Organisation d'Archer

IV-2-1) Base de connaissance

--> Une base de règles (partie opératoire) qui contient le savoir-faire lié à la modélisation des systèmes dynamiques par l'approche Bond-Graph .

--> Une base de faits (partie de données établies)qui suit l'évolution de la représentation .

Plus en détail , voici la composition de la base de connaissance en ce qui concerne Archer :

-->Une base de règles formée de sous bases :

->Une base de règles pour la construction des Bond-Graphs non simplifiés et sans causalité, à partir du domaine mécanique de dimension 1 .

->Une base de règles pour la construction des Bond-Graphs non simplifiés et sans causalité, à partir du domaine électrique .

->Une base de règles pour la simplification des Bond-Graphs .

->Une base de règles pour l'affectation des causalités .

->Une base de règles pour la construction des équations .

-->Une base de faits (partie de données établies)qui peut correspondre à :

->Un modèle utilisateur en mécanique de dimension 1 .

->Un modèle utilisateur en électrique .

->Un modèle Bond-Graph non simplifié et sans causalité .

->Un modèle Bond-Graph simplifié sans causalité .

->Un modèle Bond-Graph simplifié avec causalité .

->Un modèle équation .

IV-2-2) Un moteur d'inférences.

Nous avons choisi d'utiliser le Turbo -Prolog.

Le moteur de Prolog comme celui de Turbo Prolog est basé sur la résolution de Robinson appliquée aux clauses d'Horn , il pratique le chainage arrière avec back-tracking .

Mais pour certaines applications , il est toujours possible de modifier l'algorithme suivi pour l'effacement de buts .

Dans ce champ de vision le cut ! est un outil précieux qui permet de sauter les choix en suspens .

Aussi par exemple , dans la programmation déclarative , nous pouvons définir un jeu de macro-instructions qui peut donner naissance à une programmation procédurale au niveau des demandes de résolution des buts .

De plus Turbo Prolog s'avère être un langage trés pratique pour conférer à notre logiciel tout un ensemble de fonctions renforçant son intérêt pédagogique .

Ainsi différentes questions peuvent être posées par l'utilisation d'un même prédicat , ceci selon les différentes combinaisons d'instanciation ou non de ses arguments .

De plus,il possède des prédicats graphiques trés utiles pour la gestion de l'écran et la création de dessins.

IV-2-3) Langage d'expression des connaisssances

Il procède par l'intermédiaire d'un ensemble de prédicats à paramètres variables .

Exemple une jonction zéro de numéro 1 et portant une inertie in2 et une source de flux sf1 pourra être décrite de la manière suivante : jonction-0(1,(in2,sf1)) .

On a ainsi pratiquement un langage naturel;remarque plusieurs représentations peuvent coexister,exemple :
vecteur (Nom du vecteur, Liste des éléments)
et element-vecteur (Nom du vecteur,coordonnée,élément)
dimension-vecteur (Nom du vecteur, Nombre d'éléments) .

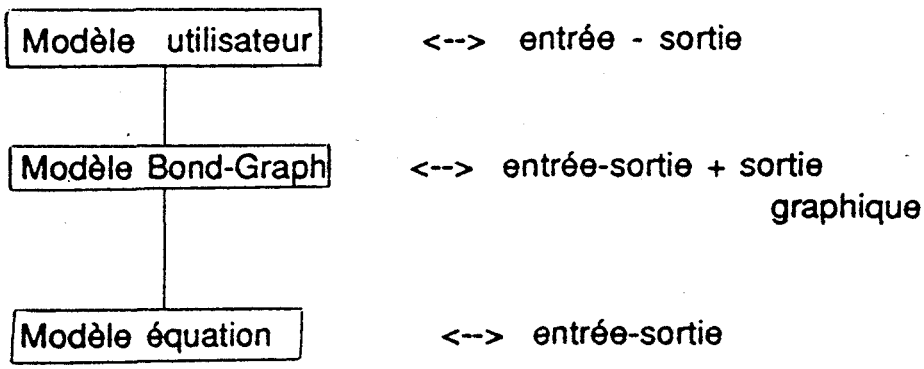
IV-2-4) Organisation fonctionnelle en modules

Nous avons une organisation fonctionnelle en modules les raisons sont conceptuelles .

Une bonne organisation permet de minimiser l'occupation mémoire , maximiser les services rendus ,etc ...

La structure modulaire permet des transformations à volonté .

D'un point de vue global, la structure d'ARCHER est présentée sur le schéma suivant,où l'on retrouve l'organisation type d'un système expert.



V) Conclusion

En conclusion ARCHER utilise :

.Un outil permettant de représenter les transferts de puissance : les Bond-Graphs.

. Des moyens de visualisation graphique.

.Une structuration fonctionnelle qui s'articule autour d'une organisation modulaire.

.Des techniques de l'intelligence artificielle
.-Base de faits :
->stockage du modèle qui pourra évoluer dynamiquement.
-Base de règles :
->stockages de quantum de savoir-faire de l'expert.

.Un langage de 5 ème génération structuré le Turbo-Prolog.

.Un matériel compatible IBM-PC.

On peut donc affirmer qu'Archer s'inscrit bien dans la lignée des systèmes experts .

De plus,il innove et il s'implante de manière complémentaire aux autres logiciels .

Actuellement il apparait comme un préprocesseur des outils de simulation existants , original dans le contexte des logiciels dédiés à la Modélisation-Simulation grâce à l'association outil Bond-Graph-Technique IA .

CHAPITRE III
LE SYSTEME EXPERT ARCHER

Chapitre III Le système expert Archer .

I) Introduction

Le système expert ARCHER utilise une structure basée sur une maquette modulaire spécialisée .

Cela le rend facilement extensible.

Dans ce chapitre , une première approche est la présentation des différents types de modèle; ils sont décrits avec leurs langages pseudo-naturels.

Ensuite, nous présentons les algorithmes qui servent dans les constructions liées à l'évolution des modèles.

Et pour terminer,le superviseur,soit l'ensemble des menus est détaillé.

Nous avons un système d'interfaçage sophistiqué visant à une adéquation parfaite entre utilisateur et machine.

II) Base de faits : "présentation des modules en langages naturel"

Archer procède à partir de modèles descriptifs de système physique .

Cela entraine de se fixer une représentation de ces systèmes , et pour cela il est nécessaire d' envisager des langages permettant leurs descriptions .

Lors des constructions des différents modèles , on utilise les procédures données dans le chapitre I , ceci nous impose un langage utilisateur différent suivant les domaines .

II-1) Modèle mécanique en dimension 1

Présentons le langage utilisateur générateur des modèles de représentation des systèmes mécaniques de dimension 1 .

En fait ce langage est constitué d'un ensemble de relations élémentaires qui sont cataloguables en trois types .

Nous avons la relation entre deux noeuds de vitesse non nulle ,
rel (< Nv1 > , < Tl > , [< Es >] , < Nv2 >) ;
puis la relation entre un noeud de vitesse non nulle et un noeud de vitesse nulle ,
bati (< Nv > , < Tl > , [< El >]) ;
et finalement la dernière relation pour le positionnement des sources ,
source (< Nv > , < Ts > , [< Es >])

Précisons les paramètres , on a :

Noeud de vitesse : < Nv > :: < ma. > I < mf. >
ma. : inertie
mf. : noeud de vitesse associé à une inertie nulle
. : nombre

Type de liaison : < Tl > :: < para > I < serie > I < transfo >
para : parallèle
serie : série
transfo : transformateur

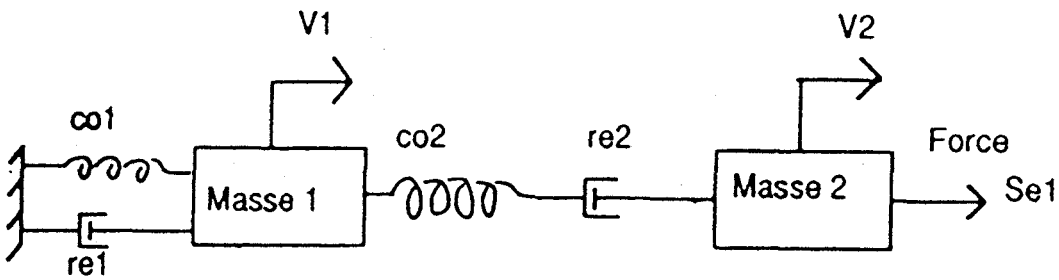
Type de source : < Ts > :: < effort > I < flux >

Elément de liaison : < El > :: < co. > I < re. > I < tf. >
(avec type de liaison)
co. : ressort
re. : amortisseur
tf. : transformateur

Elément source : < Es > :: < se. > I < sf. >
(avec type de source)
se. : source d'effort
sf. : source de flux

Remarques :
I : ou
[ ] : itérations

Considérons l'exemple suivant



Ce système s'écrit :

```
modele(mec5,me1)
rel ( ma1 , serie ( co2 , re2 ) , ma2 )
bati ( ma1, para ( co1 , re1 ) )
source ( ma2 , effort ( se1 ) )
```

Le sens d'écriture des relations donnera dans le Bond-Graph le sens des demi-flèches .

Le prédicat modèle comporte 2 arguments
(nom du modèle , type de domaine) .

Le modèle utilisateur est stocké dans la base de données sous le nom "modèle . me1" .

En annexe , sont présentés d'autres exemples de modèles mécaniques en dimension 1 .

II-2) Modèle électrique

Abordons directement la syntaxe du langage utilisateur lié à la description de système électrique .

Noeud de tension : < Nt > :: < te. >
. : nombre

Type de liaison : < Tl > :: < para > I <serie > I < tfprim > I
< tfsec > I < gene >
para : parallèle
serie : série
tfprim : primaire du transformateur
tfsec : secondaire du transformateur
gene : générateur

Elément de liaison : < El > :: < re. > I < co. > I < in. > I < tf. > I
< se. > I < sf. >
re. : résistance
co. : condensateur avec type : < serie > I < para >
in. : inductance

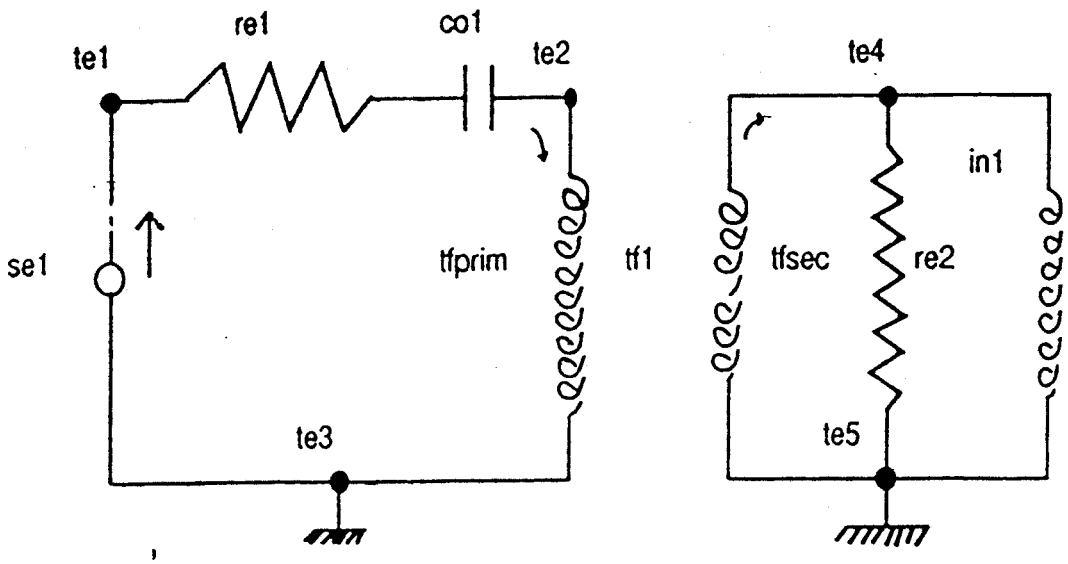
tf. :transformateur avec type : < tfprim > I < tfsec >

se. : source de tension avec type : < gene > I < serie >
sf. : source de courant I < para >

Relation entre noeuds de tension :
rel (< Nt1 > , < Tl > , [< El >] , < Nt2 >)

Les points de masse :
point_de_masse (< Nt >)

Soit l'exemple suivant :



Ce système s'écrit :

```
rel ( te3 , gene ( se1 ) , te1 )
rel ( te1 , serie ( re1 , co1 ) , te2 )
rel ( te2 , tfprim ( tf1 ) , te3 )
rel ( te5 , tfsec (tf1 ) , te4 )
rel ( te4 , para ( re2 , in1 ) , te5 )
point_de_masse ( te3 )
point_de_masse ( te5 )
```

Le sens à retenir pour l'écriture des relations est celui défini pour le sens du courant et donne le sens de l'orientation des demi-flèches .

Cet exemple est stocké dans la base de données sous l'appelation "modele.ele" .

En annexe , sont présentés d'autres exemples de modèles électriques .

II-3) Modèle Bond-Graph

On distingue 3 étapes dans sa construction ce qui nous donne 3 classes de modèle Bond-Graph .

Soit :
."bgs" : Bond-Graph non simplifié sans causalité .
."bgp" : Bond-Graph simplifié sans causalité .
."bgc" : Bond-Graph simplifié avec causalités .

Il y a ajout et suppression de relation au fil de l'évolution du modèle .

Et nous avons le langage syntaxique suivant (type "bgc")
Num : numéro identifiant la jonction
Composant : < Cp > :: < in. > I < co. > I < re. > I
< se. > I < sf. >
in. : élément inertiel
co. : élément capacitif
re. : élément amortisseur
se. : élément source d'effort
sf. : élément source de flux

Les jonctions :

jonction_1 (< Num > , [< Cp >])
jonction_0 (< Num > , [< Cp >])
jonction_tf (< Num >)
jonction_gy (< Num >)

Les liens

lien_01(<Num>,<Num>) lien_10(<Num>,<Num>)
lien_0tf(<Num>,<Num>) lien_tf0(<Num>,<Num>)
lien_1tf(<Num>,<Num>) lien_tf1(<Num>,<Num>)
lien_0gy(<Num>,<Num>) lien_gy0(<Num>,<Num>)
lien_1gy(<Num>,<Num>) lien_gy1(<Num>,<Num>)
lien_11(<Num>,<Num>) lien_00(<Num>,<Num>)

Les causalités et le numérotage

numero_effort (< Num > , < Elt1 > , <Elt2 >)
causalite_derivée (< Elt >)
<Elt> :: <cp>I<un.>I<zer.>I<tf.>I<gy.>
nombre_de_liens (< Num >)
Des exemples sont donnés en annexe .

MODELE BOND-GRAPH SANS CAUSALITE . NON SIMPLIFIE
NOM : mec5

```
modele(mec5,bgs)
jonction_1(1,(in2,se1))
jonction_1(2,(in1,co1,re1))
jonction_0(1,(co2,re2))
lien_01(1,1)
lien_10(2,1)
```

MODELE BOND-GRAPH SANS CAUSALITE SIMPLIFIE .
NOM : mec5

```
modele(mec5,bgp)
jonction_1(1,(in2,se1))
jonction_1(2,(in1,co1,re1))
jonction_0(1,(co2,re2))
lien_01(1,1)
lien_10(2,1)
```

MODELE BOND-GRAPH AVEC CAUSALITES .
NOM : mec5

```
modele(mec5,bgc)
jonction_1(1,(in2,se1))
jonction_1(2,(in1,co1,re1))
jonction_0(1,(co2,re2))
lien_01(1,1)
lien_10(2,1)
numero_effort(1,se1,un1)
numero_effort(2,un1,in2)
numero_effort(3,zer1,un1)
numero_effort(4,un2,in1)
numero_effort(5,co1,un2)
numero_effort(6,re1,un2)
numero_effort(7,zer1,un2)
numero_effort(8,co2,zer1)
numero_effort(9,zer1,re2)
nombre_de_liens(9) .
```

II-4) Modèle équation

C'est le dernier type de modèle envisagé et pour sa présentation 2 formes de syntaxe sont prises en compte .

La description sous syntaxe proche de celle de Matlab est mise au point à partir des données sous la 1 ère forme que voici .

Syntaxe du langage descriptif des équations :

Les dimensions :

dimension_vecteur(<Nom_du_vecteur>,<Nombre_de_ligne>)
dimension_matrice(<Nom_de_matrice>,
<Nombre_de_lignes>,<Nombre_de_colonnes>)

Les vecteurs :

element_vecteur(<Nom_du_vecteur>,<Numéro_de_ligne>,
<Valeur>)
element_matrice(<Nom_de_matrice>,<Numero_de_ligne>,
<Numero_de_colonnes>,<Valeur>)

Et nous avons l'exemple mec5.equ sous les 2 formes (d'autres exemples sont donnés en annexe).

```
-------------------------
 VECTEURS ET MATRICES .
 NOM : mec5
-------------------------

  modele(mec5,equ)
  dimension_vecteur(Xi',4)
  dimension_vecteur(Din,2)
  dimension_vecteur(sortie_matrice_S,6)
  dimension_vecteur(Zi,4)
  dimension_vecteur(Xd',0)
  dimension_vecteur(Zd,0)
  dimension_vecteur(Dout,2)
  dimension_vecteur(source_U,1)
  dimension_vecteur(entree_matrice_S,7)
  dimensions_matrice(L,2,2).
  dimensions_matrice(Fi,4,4)
  dimensions_matrice(Fd,0,0)
  dimensions_matrice(S,6,7)
  element_vecteur(Xi',1,e2)
  element_vecteur(Xi',2,e4)
  element_vecteur(Xi',3,f5)
  element_vecteur(Xi',4,f8)
  element_vecteur(Din,1,e9)
  element_vecteur(Din,2,f6)
  element_vecteur(sortie_matrice_S,1,e2)
  element_vecteur(sortie_matrice_S,2,e4)
  element_vecteur(sortie_matrice_S,3,f5)
  element_vecteur(sortie_matrice_S,4,f8)
  element_vecteur(sortie_matrice_S,5,e9)
  element_vecteur(sortie_matrice_S,6,f6)
  element_vecteur(Zi,1,f2)
  element_vecteur(Zi,2,f4)
  element_vecteur(Zi,3,e5)
  element_vecteur(Zi,4,e8)
  element_vecteur(Dout,1,f9)
  element_vecteur(Dout,2,e6)
  element_vecteur(source_U,1,e1)
  element_vecteur(entree_matrice_S,1,f2)
  element_vecteur(entree_matrice_S,2,f4)
  element_vecteur(entree_matrice_S,3,e5)
  element_vecteur(entree_matrice_S,4,e8)
  element_vecteur(entree_matrice_S,5,f9)
  element_vecteur(entree_matrice_S,6,e6)
  element_vecteur(entree_matrice_S,7,e1)
  element_matrice(L,1,1,1/re2)
  element_matrice(L,2,2,re1)
  element_matrice(Fi,1,1,1/in2)
  element_matrice(Fi,2,2,1/in1)
  element_matrice(Fi,3,3,1/co1)
  element_matrice(Fi,4,4,1/co2)
  element_matrice(S,1,4,+1)
  element_matrice(S,1,7,+1)
  element_matrice(S,2,3,-1)
  element_matrice(S,2,6,-1)
  element_matrice(S,2,4,-1)
  element_matrice(S,3,2,+1)
  element_matrice(S,4,1,-1)
  element_matrice(S,4,2,+1)
  element_matrice(S,4,5,-1)
  element_matrice(S,5;4,+1)
  element_matrice(S,6,2,+1)
```

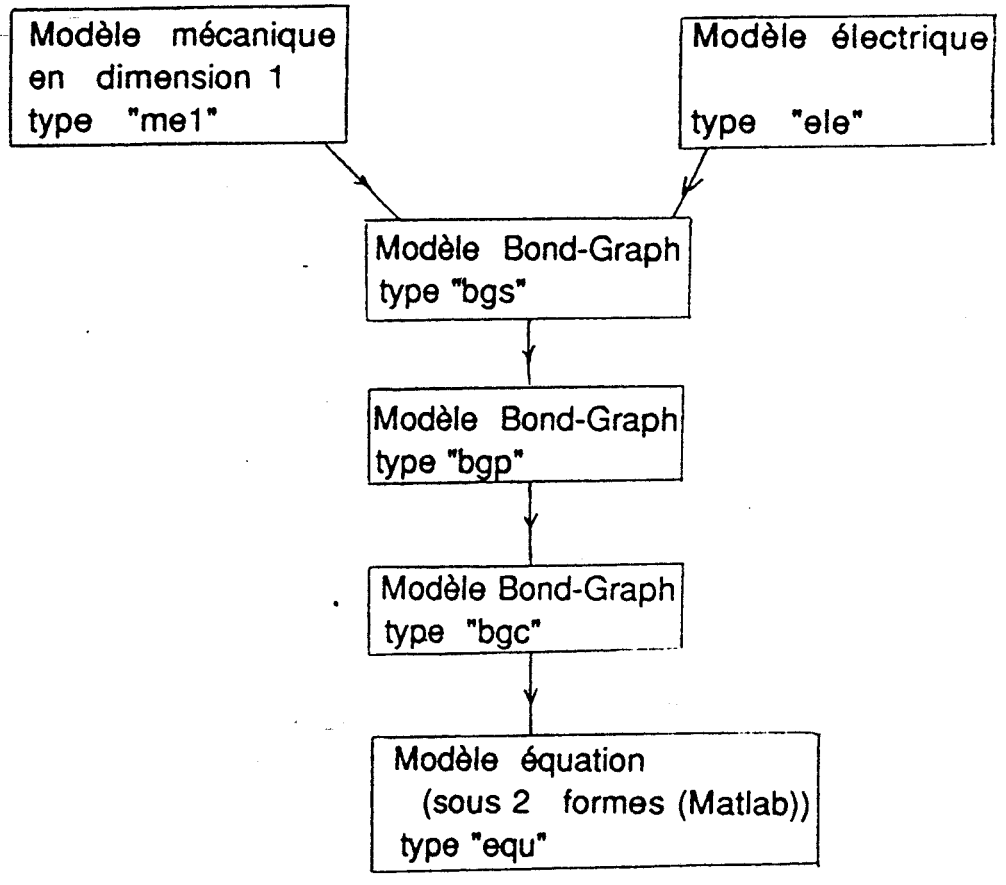
SYNTAXE SOUS FORME MATLAB

VECTEURS ET MATRICES .
NOM : mec5

```
Vecteur Xi' :
[ e2 ; e4 ; f5 ; f8 ]
Vecteur Din :
[ e9 ; f6 ]
Vecteur de sortie de la matrice S :
[ e2 ; e4 ; f5 ; f8 ; e9 ; f6 ]
Vecteur Zi :
[ f2 ; f4 ; e5 ; e8 ]
Vecteur Zd :
[ ]
Vecteur Xd' :
[ ]
Vecteur Dout :
[ f9 ; e6 ]
Vecteur source U :
[ e1 ]
Vecteur d'entrée de la matrice S :
[ f2 ; f4 ; e5 ; e8 ; f9 ; e6 ; e1 ]
Matrice L telle que Dout = L Din :
[ 1/re2 0 ;
  0 re1 ]
Matrice Fi telle que Zi = Fi Xi :
[ 1/in2 0 0 0 ;
  0 1/in1 0 0 ;
  0 0 1/co1 0 ;
  0 0 0 1/co2 ]
Matrice Fd telle que Zd = Fd Xd :
[ ]
Matrice S :
[ 0 0 0 +1 0 0 +1 ;
  0 0 -1 -1 0 -1 0 ;
  0 +1 0 0 0 0 0 ;
  -1 +1 0 0 -1 0 0 ;
  0 0 0 +1 0 0 0 ;
  0 +1 0 0 0 0 0 ]
```

II-5) Evolution des langages descriptifs

En récapitulatif , nous avons :



Soit 7 langages pour décrire ces 6 modèles .

Remarque : chacun de ces langages est associé à un analyseur syntaxique qui permet d'avoir une vérification de la forme .

Attention les analyseurs syntaxiques ne prennent pas en compte les erreurs sémantiques .

Il serait peut être interessant de recenser une liste d'erreurs sémantiques .

Ceci dit la sémantique est prise en considération lorsque l'on s'attache par exemple à la détection des causalités dérivées .

III) Base de règles : "les algorithmes"

Ici , nous présentons uniquement les algorithmes de construction ; c'est à dire tout ce qui concerne l'évolution des modèles avec leur changement de type .

Un menu lié uniquement à la commande de cette partie a été développé .

L'aspect modulaire de cette partie permet d'intégrer sans problème d'autres éléments fonctionnels .

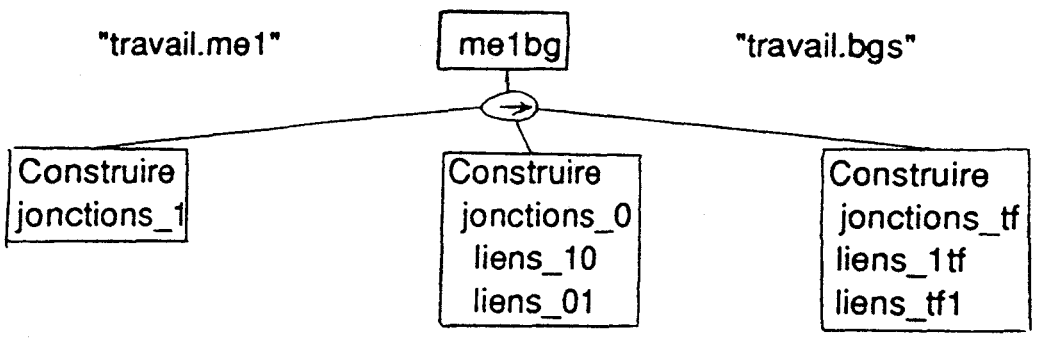
III-1) Construction du Bond-Graph sans causalité à partir d'un modèle mécanique de dimension 1 .

Les faits à traiter sont écrits sous forme de prédicats associés au langage utilisateur mécanique 1.
Nous le notons LUM1(Langage utilisateur mécanique1)

*LUM1

rel (nom , nom , liste , nom)
bati (nom , nom , liste)
source (nom , nom , liste)

* Algorithme



Et plus précisemment , en détaillant les opérations :
- La construction des jonctions 1 s'opère pour chaque noeud de vitesse , soit pour chaque point de masse fictive ou réelle .

La première opération consiste à dresser la liste de ces noeuds de vitesse de masse fictive ou réelle , ensuite pour chaque élément de la liste , une jonction_1 est créée , elle est marquée par un numéro .

Une table de correspondance entre les numéros et les noeuds de vitesse de masse fictive ou réelle qui se correspondent est alors dressée , elle servira par la suite .

Lorsque une jonction_1 est construite , il y a aussi recherche des éléments qui y sont attachés ; donnons simplement la liste des types d'éléments trouvables :

- élément inertiel "in"
- élément source "sf" , "se"

Remarque : il faut considérer les éléments de liaison en parallèle ou l'unique élément en série qui relie un noeud de vitesse au bati pour former une nouvelle liste .

- La 2 ème étape est la construction des jonctions 0 et des liens associés .

Pour chaque relation entre noeud de vitesse , il va y avoir apparition de jonctions 0 et des liens qui les accompagnent .

Ces jonctions 0 seront aussi numérotées .
On distingue deux cas :

. Type de liaison serie -->
On a alors un effort commun et une seule jonction 0 suffit; tous les éléments de la liaison composent , de ce fait , la liste des éléments liés à la jonction 0 .
Puis on ajoute le lien 10 et le lien 01 , mais pour cela ,il faut rechercher le numéro des jonctions1 qui sont associés au noeud de vitesse mis en jeu dans la relation .

. Type de liaison para -->
On a alors un flux commun et pour chaque élément de liaison , il faut créer une jonction 0 à laquelle il sera lié .
Pour chaque création de jonction 0 les liens 10 et 01 seront aussi à générer .

Pour les relations entre bati et noeuds de vitesse , il y aura aussi création de jonctions 0 dans le cas où l'on a des efforts communs , c'est à dire si il y a plus d'un élément en série reliant un noeud de vitesse au bati .
Pour chaque création de jonction 0 , les liens 01 sont alors générés .

La 3 ème étape consiste en la création des jonctions tf et des liens 1tf et tf1 .
Pour cela on crée les jonctions tf et les liens associés pour chaque relation ayant le type de liaison transfo .

La numérotation des jonctions tf est déduite directement des numéros respectifs des tf donnés dans les relations .

La 4 ème étape consiste en l'initialisation qui fait disparaitre toute relation subsistante du modèle utilisateur ainsi que la table de correspondance qui aura servi durant certaines recherches .

Le modèle Bond-Graph sans causalité alors obtenu se présente sous la forme que nous notons BG (Bond-Graph).

jonction_1(Numéro de la jonction,Liste des éléments attachés)
jonction_0(nombre,liste)
jonction_tf(Numéro de la jonction)
jonction_gy(nombre)
lien_01(Numéro de la 1 ère jonction,Numéro de la 2 ème jonction)
lien_10(nombre,nombre)
lien_0tf(nombre,nombre)
lien_tf0(nombre,nombre)
lien_1tf(nombre,nombre)
lien_tf1(nombre,nombre)
lien_0gy(nombre,nombre)
lien_gy0(nombre,nombre)
lien_1gy(nombre,nombre)
lien_gy1(nombre,nombre)
lien_11(nombre,nombre)
lien_00(nombre,nombre)

Dans la dernière version réalisée ce bloc est lui sauvé dans un fichier de suffixe "bgs"

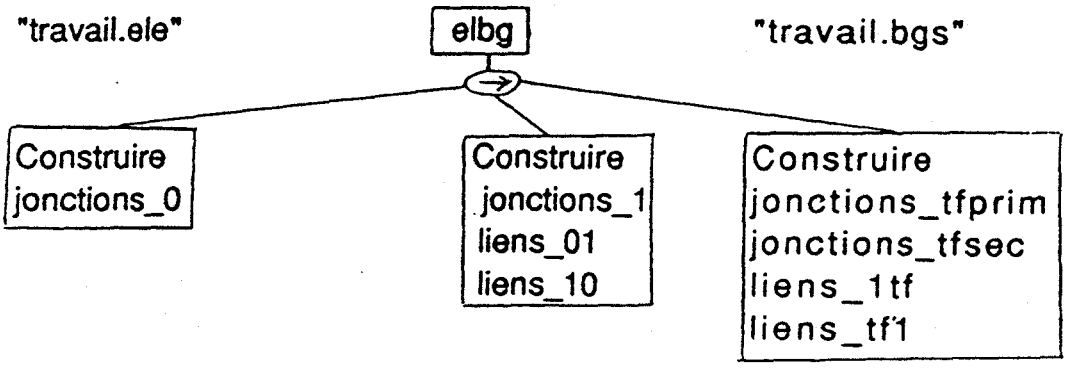
III-2) Construction du Bond-Graph sans causalité à partir d'un modèle électrique .

L'ensemble des prédicats associés au modèle utilisateur électrique est noté LUE (Langage utilisateur électrique).

* LUE

rel (nom,nom,liste,nom)
point_de_masse(nom)

* L'algorithme de construction est le dual de celui présenté dans le cas mécanique 1.



Détaillons les différentes étapes

1) Construction des jonctions 0
Pour chaque élément de tension de la liste une jonction 0 est créée,elle est marquée par le numéro extrait du noeud de tension.

2) Construction des jonctions1 et des liens 01 et 10
Pour chaque relation entre noeuds de tension ,il y a apparition de jonctions1 et des liens qui les accompagnent , ces jonctions 1 seront numérotées.
On distingue quatre cas :
. Type de liaison série
On a alors un flux commun et une seule jonction1 suffit;tous les éléments de la liaison composent,de ce fait,la liste des éléments liés à la jonction 1.
Puis on ajoute le lien 01et le lien10 .
.Type de liaison para
On a alors un effort commun et pour chaque élément de liaison il faut créer une jonction1 à laquelle il sera lié.
Pour chaque création de jonction1 les liens 01 et10 seront aussi générer.
.Type de liaison gene
On a alors une jonction 1 sur laquelle est porté l'élément source .
Puis on ajoute le lien 01 et le lien 10 .
.Type de liaisons tfprim et tfsec
On crée alors deux jonctions1, une pour le primaire l'autre pour le secondaire , et chacune d'elles les liens 01 et 10 qui les accompagnent .
La table "nom-nombre" est mise à jour de façon à repérer le couple de jonction-1 vis à vis du tf.

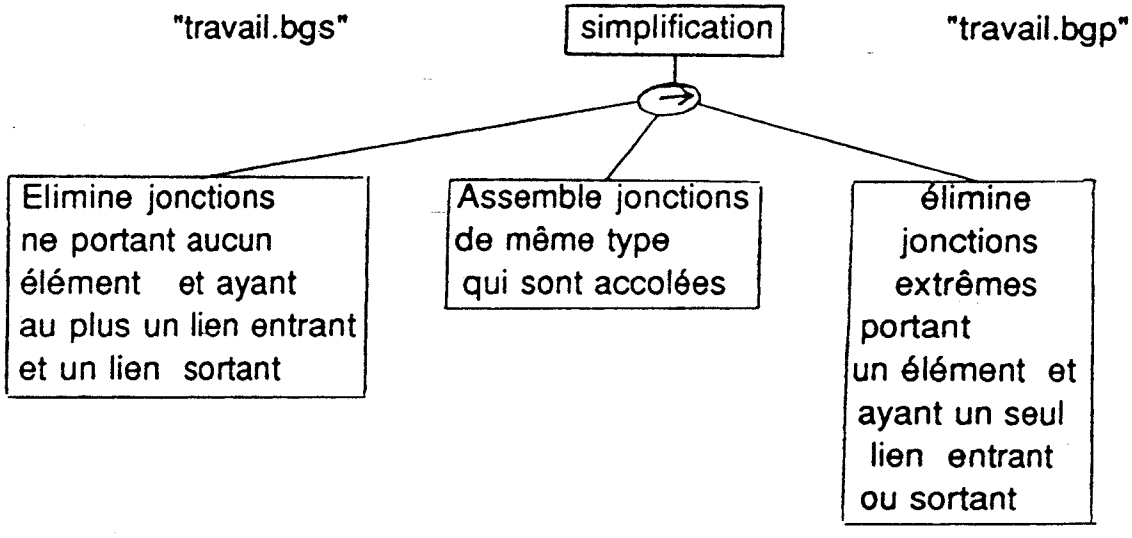
3) Construction des jonctions tf et des liens 1tf et tf1
On crée les jonctions tf et les liens associés pour chaque couple de relation ayant les types de liaison tfprim et tfsec . La numérotation des jonctions tf est déduite directement des numéros respectifs des tf donnés dans les relations.

4) Simplifications liées aux points de masse
Les jonctions 0 associées au noeuds de tensions choisies comme point de masse sont éliminées ainsi que les liens qui y sont attachés.
L'ensemble de faits obtenus à cette étape se présente sous la forme BG , comme en mécanique 1.

III-3) Simplification des Bond-Graphs sans causalité

L'ensemble de faits à traiter apparait dans chaque cas sous la forme BG.

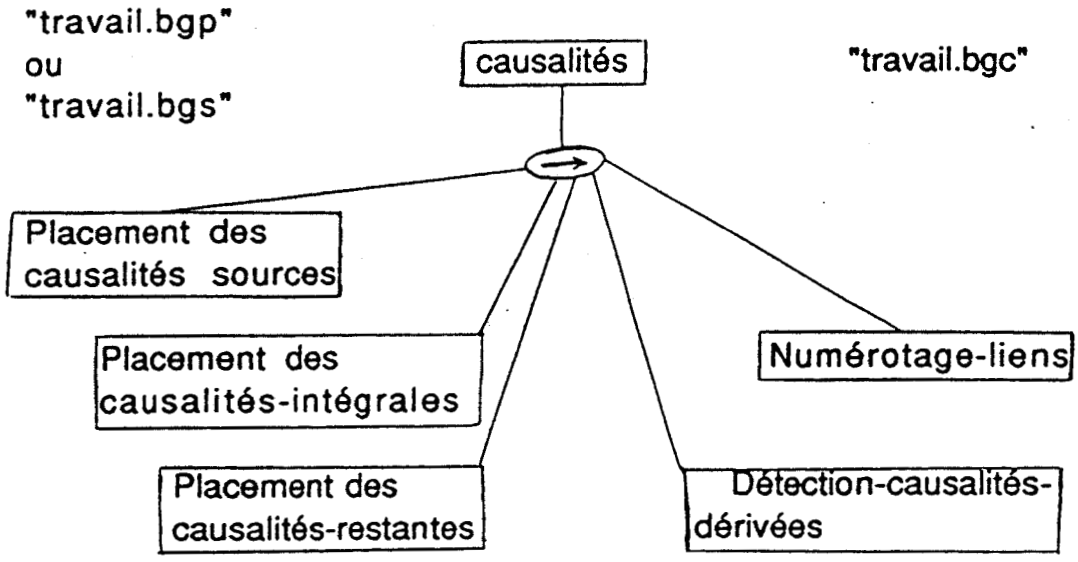
* L'algorithme de simplification permet d'éliminer les éléments inutiles.



III-4) Construction des causalités

* Modèles Bond-Graph sans causalité
L'ensemble des faits à traiter apparait sous forme BG avec un nombre minimal de faits obtenus aprés simplification.

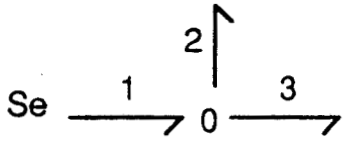
* L'algorithme développé suit la procédure d'affectation de la causalité présentée dans le premier chapitre.

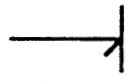


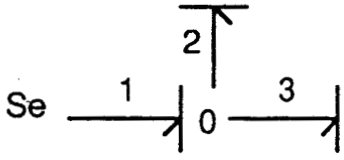
Détail de l'algorithme

On affecte d'abord les causalités(obligatoires)aux sources et on cherche à répercuter sur les jonctions et les liens proches les conséquences de cette affectation, en respectant les restrictions liées aux jonctions.

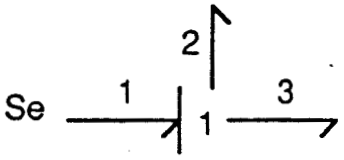
Ainsi dans le 1er cas



l'affectation de la causalité sur Se (Se ) implique directement la causalité sur les liens 2 et 3 tels que

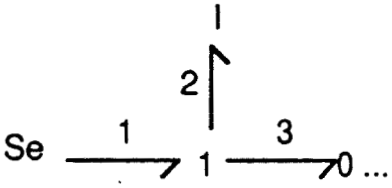


alors que aucune conclusion n'est possible immédiatement dans le cas



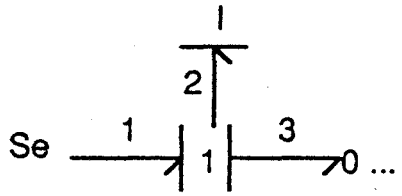
On positionne les causalités intégrales et celles qui en découlent directement.

Considérons à titre d'exemple le cas suivant

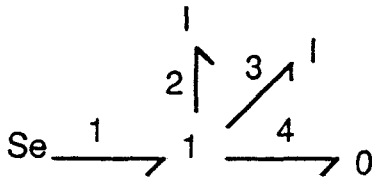


L'étape 1 affecte la causalité à Se, l'étape 2 affecte la causalité à I, et directement la causalité au lien 3.

Dans ce cas il n'y a pas de conflit de causalité, et on obtient

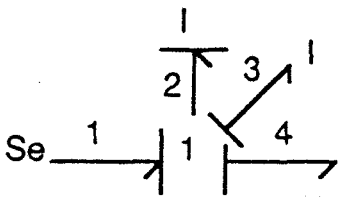


Par contre, le modèle Bond-Graph suivant



est traité tout d'abord pour Se, puis I2 et les liens 3 et4 sont alors assignés.

On obtient

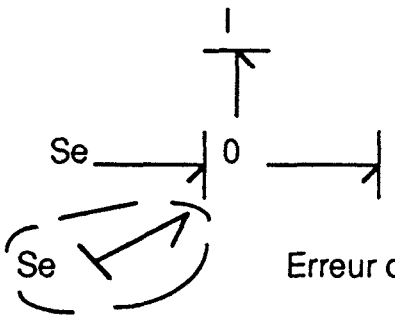


d'où apparition d'une causalité dérivée sur I3.

Puis on place les causalités restantes (sur les R), on détecte les causalités dérivées et les conflits ou erreurs de causalité.

Ceux-ci sont en nombre trés limité, à cause des procédures utilisées.

Ils apparaissent par exemple quand plusieurs sources sont attachées à une même jonction, par exemple



Erreur de causalité

On numérote les liens, dans l'ordre d'affectation de la causalité.

Le fait effort(Nom1,Nom2) indique que l'effort va de Nom1 vers Nom2 ; la causalité se trouve donc placer sur Nom2 .

Aprés le numérotage des liens on retrouve la causalité sous le prédicat "numero-effort(Nombre,Nom1,Nom2)" , la

seule diférence avec le prédicat précédent est le nombre placé devant pour indiquer le numéro du lien .

Un prédicat signale les causalités dérivées rencontrées "causalité-dérivée(Nom)" .

Un autre prédicat donne le nombre de liens aprés le numérotage "nombre-de-liens(Nombre)" .

A cette étape, les faits à traiter se présentent sous la forme BG, à laquelle sont ajoutés les faits

effort(nom,nom)
numero-effort(nombre,nom,nom)
causalité-dérivée(nom)
erreur-causalite(nom)
nombre-de-liens(nombre)

Cet ensemble de faits(BG+Causalité) est noté BGC (Bond_Graph Causal).

Remarque :pour placer les causalités un certain nombre de prédicats utilitaires spécifiques ont été définis.

Citons :

.Extrait d'une liste d'éléments lié à une jonction i,la liste des éléments dont les liens avec i sont sans causalité.

.Trouve la liste des éléments(composants et jonctions) qui sont reliés à la jonction i par des liens sans causalité.

.Vérifie si la causalité du lien X-Y est attribuée.

.Comptabilise le nombre d'efforts entrant sur i parmi les causalités déjà attribuées.

.Comptabilise le nombre d'efforts sortant sur i parmi les causalités déjà attribuées.

.Compte les causalités non attribuées sur une jonction i.

.Compte les causalités non attribuées sur les liens entre une jonction i et une liste d'éléments attachés.

.Place des efforts entrants(respectivement sortants).

.Vérifie que toutes les causalités sauf une ont été attribuées sur les liens de cette jonction-1 et qu'elles sont entrantes , ainsi il ne reste qu'une causalité sortante de la jonction-1 à attribuer.

.Il y a eu attribution d'une causalité sortante sur une jonction-1,toutes ses causalités non attribuées seront positionnées de façon entrante.

.Vérifie que toutes les causalités sauf une ont été attribuées sur les liens de cette jonction-0 et qu'elles sont sortantes , ainsi il ne reste qu'une causalité entrante sur la jonction-0 à attribuer.

.Il y a eu attribution d'une causalité entrante sur une jonction-0,toutes ses causalités non attribuées seront positionnées de façon sortante.

.Sert à poursuivre le placement des causalités sur une jonction i , si nécessaire.

III-5) Construction de l'équation d'état associé au modèle

L'ensemble de faits à traiter est sous la forme BGC.

* L'algorithme développé se compose des étapes suivantes:

. Recherche des flux et des efforts en tenant compte des égalités flux aux jonctions 1 et efforts aux jonctions 0 .

. Elaboration du vecteur de sortie de la matrice S ainsi que les vecteurs X'i et Din .

. Elaboration du vecteur d'entrée de la matrice S ainsi que des vecteurs : Zi , X'd , Zd , Dout et Source-U .

. Recherche de la dimension de chaque vecteur .

. Recherche des dimensions de chaque matrice .

. Partitionnement des vecteurs .

. Elaboration des éléments de la matrice L .

. Elaboration des éléments de la matrice Fi .

. Elaboration des éléments de la matrice Fd .

. Elaboration des éléments de la matrice S .

Les fichiers utilisés à cette étape du traitement sont "travail.bgc" et "travail.equ" , les faits obtenus se présentent sous la forme suivante:

```
numero_causalite_effort_flux
( nombre , nom , nom , nom , nom )
vecteur ( chaine , liste )
relation_e ( nom , nom , nom )
relation_f ( nom , nom , nom )
fonction_e ( nom , nom , chaine )
fonction_f ( nom , nom , chaine )

dimension_vecteur ( chaine , nombre )
element_vecteur ( chaine , nombre , chaine )
dimensions_matrice ( chaine , nombre , nombre )
element_matrice ( chaine , nombre , nombre , chaine )
```

IV) Le superviseur : "les menus"

- Archer est muni d'un ensemble d'outil d'édition . ("Archer.prj")

Le menu principal se compose de :

Editeurs
Coupleur
Système de construction
Graphisme
Chargement d'une session
Sauvegarde de la session
Nettoyage complet de la session
Quitter

L'appels d'un élément du menu renvoie au module correspondant :

Editeurs --> "edit.prj"
Coupleur --> "coupl.prj"
Systèmes de construction --> "const.prj"
Graphisme --> graph.pro

Une session regroupe une collection de modèles construits successivement .

Exemple :

modèle.me1
modèle.bgs
modèle.bgp
modèle.bgc
modèle.equ

La trace de l'ensemble des constructions d'un modèle peut être ainsi conservée .

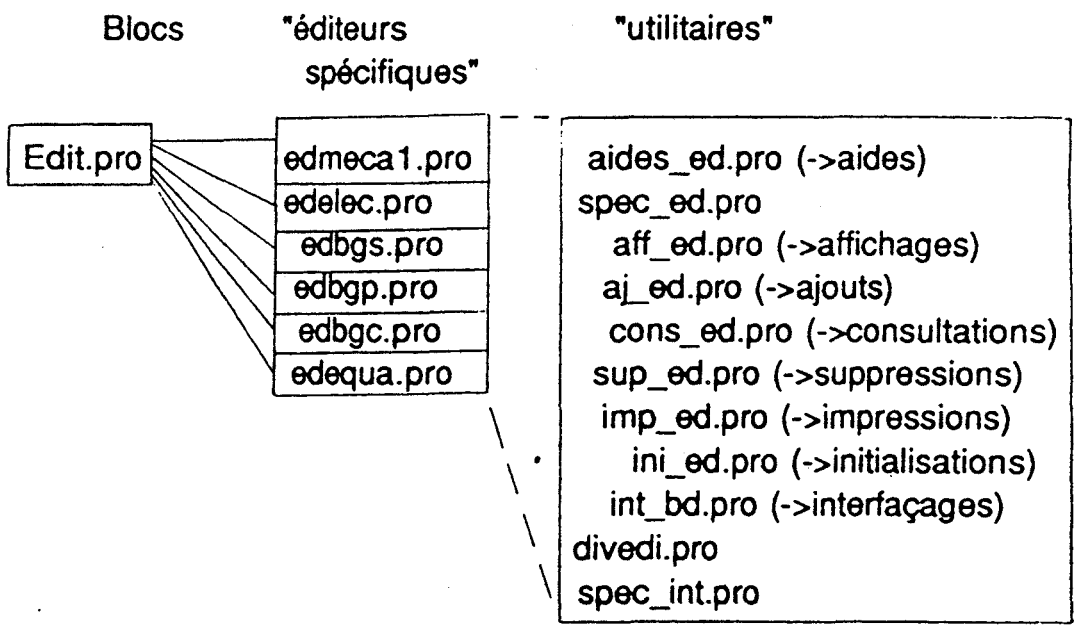
A titre d'exemple, détaillons deux modules :

* Editions ("edit.prj")

Le menu du module edit est :

Editeur des modèles mécaniques en dimension 1
Editeur des modèles électriques
Editeur des modèles Bond-Graphs non simplifiés et sans causalité
Editeur des modèles Bond-Graphs simplifiés et sans causalité
Editeur des modèles Bond-Graphs avec causalités
Editeur lié aux équations
Sortie du menu

L'architecture du module est la suivante :



* Constructions ("const.prj")

Le menu lié à l'outil de construction est:

Mise en équation directe
Modèle utilisateur --> Modèle Bond-Graph avec causalités
Modèle Bond-Graph avec causalités --> Equations
Modèle utilisateur --> Modèle Bond-Graph sans causalités
Simplification du Bond-Graph
Placement des causalités
Sortie du menu

Cet outil de construction est opérationnel grâce à un ensemble de blocs fonctionnels .

me1bg --> modèle mécanique de dimension 1 en Bond-Graph sans causalité
elbg --> modèle électrique en Bond-Graph sans causalité
simp --> simplification du Bond-Graph sans causalité
caus --> placement des causalités
equa --> mise en équation (vecteurs - matrices)

V) Conclusion

Nous venons de présenter le corps principal d'ARCHER.

Comme nous avons pu le constater, le langage utilisateur pour représenter les systèmes physiques est trés simple et constitue pratiquement un langage naturel.

A titre d'évolution du langage,des généralisations peuvent être émises.

Par exemple, au lieu d'avoir différents types de prédicat selon la jonction, on peut avoir différents arguments selon le type de la jonction; jonction-1(1,in1) devient la jonction ("un",1,in1), ce qui revient à créer un nouveau prédicat généralisé, jonction(Type de la jonction,Numéro de la jonction, Liste d'éléments attachés à la jonction).

Ceci suppose en arrière plan une modification des algorithmes d'analyse syntaxique et de construction mais celle-ci n'est pas nécessaire si cette forme vient s'ajouter en complément.

La souplesse du système permet une transformation des données ce qui facilite le coté progiciel d'ARCHER.

En effet les interfaces de transformation des données sont faciles à concevoir.

Dans le chapitre suivant nous abordons les problèmes de couplage.

En effet, les modèles sous forme langage utilisateur sont assez long à écrire.

Il peut être intéressant de prédéfinir des modèles, stockés dans une base de données et de les associer.

Cette démarche va demander une extension du langage d'ARCHER.

CHAPITRE IV

SPECIFICATIONS DE LA PARTIE COUPLAGE

Chapitre IV

Spécifications de la partie couplage .

I) Introduction

Nous rappelons que le couplage est la possibilité d'assembler des sous-modèles prédéfinis et stockés dans une base de données.

Les modèles peuvent être de différentes espèces ; nous avons la possibilité de constituer des blocs de modèles utilisateurs ou des blocs de modèles Bond-Graph .

Il apparait donc un choix sur la nature des blocs à coupler .

Dans la suite de ce chapitre , c'est au niveau des blocs utilisateurs que les couplages sont opérés .

La première étape dans les spécifications de la partie couplage consiste en la description syntaxique des modèle à coupler et des relations opérant le couplage .

Lors de cette phase de présentation , il y a la construction de nouveaux éditeurs spécialisés pour réaliser les modèles à coupler et le système de relations opérant le couplage .

La deuxième étape proposée dans ce chapitre est l'algorithme qui traite l'ensemble des modèles et des relations de façon à obtenir le couplage .

Et l'ensemble se termine par un exemple où l'algorithme est mis en oeuvre .

II) Modèle de couplage

II-1) Modèles à coupler

La première étape consiste à construire des modèles couplables à partir de modèles de type LUM ou LUE définis dans le précédent chapitre.

Ceci impose l'introduction d'un marquage des points de couplage.

Dans le cas d'un système mécanique en dimension 1, le modèle stocké dans un fichier "_.me1" devient par l'éditeur edcm1.pro le modèle "_.cm1" obtenu en ajoutant à la syntaxe du modèle "_.me1"

{ rel (< Nv1 > , < Tl > , [< El >] , < Nv2 >)
{ bati (< Nv > , < Tl > , [< El >])
{ source (< Nv > , < Ts > , [< Es >])

le prédicat marquage (<Mq>,<Nv>).

Dés lors il est possible de se constituer une collection de modèles plus ou moins compliqués à coupler .

De même que cela vient d'être présenté au niveau modèle utilisateur mécanique en dimension 1 ; le même type d'approche sera réalisé pour les systèmes électriques .

A partir de la syntaxe d'un "_ . ele"

{ rel (< Nt1 > , < Tl > , [< El >] , < Nt2 >)
{ point_de_masse (< Nt >)

on obtient la syntaxe d'un "_ . cel" en ajoutant le prédicat

marquage (< Mq > , < Nt >)

II-2) Modèles de système de couplage

La construction des modèles de couplage se fait sous l'éditeur edcpl.pro

Syntaxe d'un "_ . cpl"
Un nouveau prédicat est introduit.
Il s'agit
bloc (< Ib > , < Nfb > , < Tfb >)
avec
< Ib > :: identificateur de bloc
< Nfb > :: nom du fichier bloc
< Tfb > :: type du fichier bloc :: < cm1 > I < cel >
qui sert à repérer tous les modules utilisés .

De plus, des prédicats faisant intervenir des relations avec des blocs sont définis.
Ils sont différents suivant le domaine concerné:

Domaine mécanique

rel_pb (< Nv1 > , < Tl > , [< El >] , < Ib2 > , < Mq2 >)
rel_bp (< Ib1 > , < Mq1 > , < Tl > , [< El >] , < Nv2 >)
rel_bb (< Ib1 > , < Mq1 > , < Tl > , [< El >] , < Ib2 > , < Mq2 >)
bati_b (< Ib > , < Mq > , < Tl > , [< El >])
source_b (< Ib > , < Mq > , < Ts > , [< Es >])

Domaine électrique

rel_pb (< Nt1 > , < Tl > , [< El >] , < Ib2 > , < Mq2 >)
rel_bp (< Ib1 > , < Mq1 > , < Tl > , [< El >] , < Nt2 >)
rel_bb (< Ib1 > , < Mq1 > , < Tl > , [< El >] , < Ib2 > , < Mq2 >)
point_de_masse_b (< Ib > , < mq >)

Le file qui contient toutes ces relations a le suffixe "cpl"

II-3) Editeurs de couplage

Les éditeurs que nous avons développés sont:

Editeur des modèles mécaniques en dimension 1 couplables .

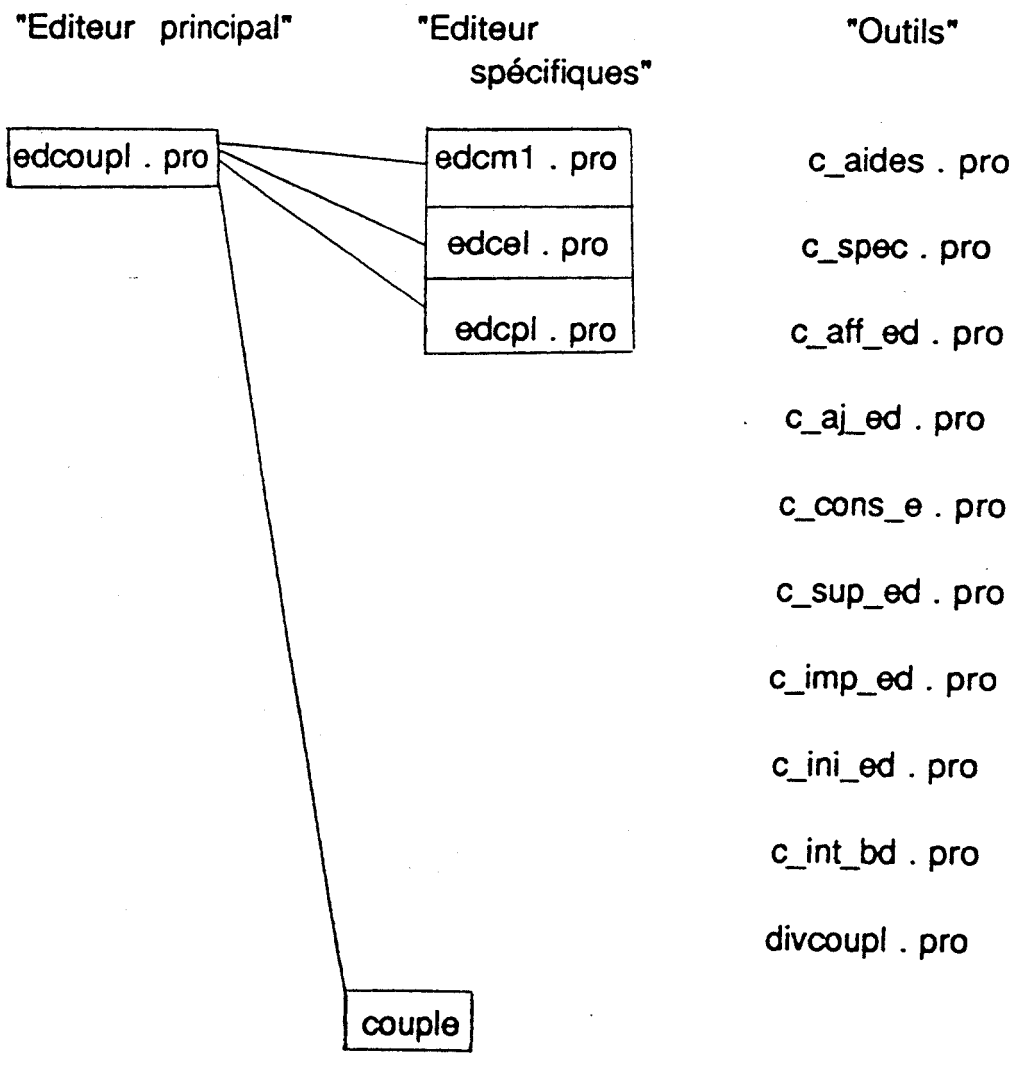
Editeur des modèles électriques couplables .

Editeur des modèles de couplage .

Coupleur .

Sortie menu .

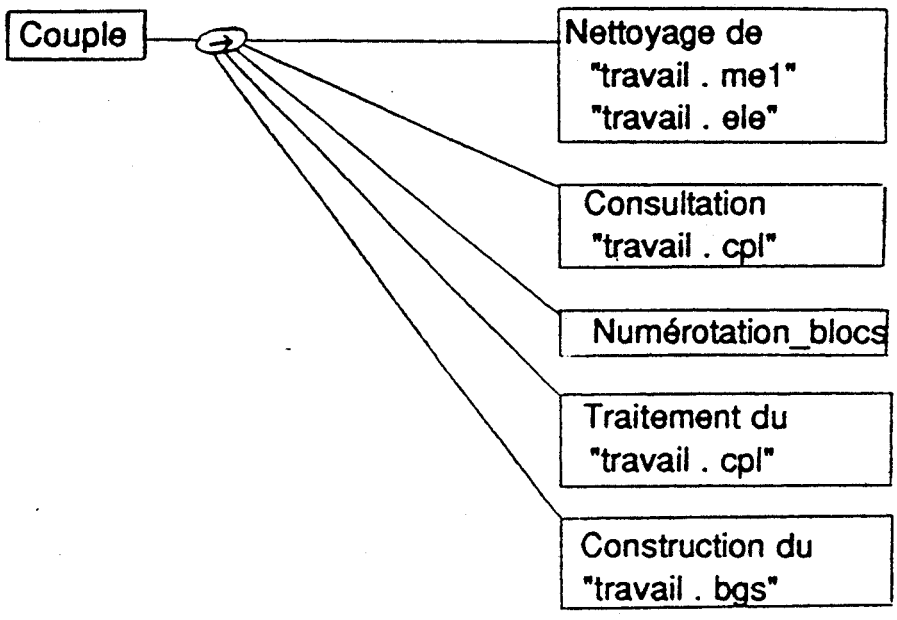
Les blocs liés au menu sont:



Une fois les éditeurs spécifiques réalisés : reste à réaliser la partie algorithmique traitée par le module couple correspondant au coupleur .

III) Algorithmes de couplage

Et nous avons les modules de construction lié à l'algorithme de couplage



Algorithmes de couplage

-> Traitement des "_ . cm1" et "_ . cel"

On lit "travail . cpl" de façon à pointer les blocs (< lb > , < Nom_file > , < Type_file >) on concatène : < Nom_file > . < Type_file > et on balaie les "_ . cm1" et les "_ . cel" .

Ils sont renumérotés de 100 en 100 et sauvegardés selon leur type dans "travail . me1" ou dans "travail . ele" .

Les relations marquage (< Mq > , < Nv >)
marquage (< Mq > , < Nt >)
vont être renumérotées et sauvegardées dans " travail . cpl" sous la forme :
bloc_marquage (< lb > , < Marque > , < Noeud >)

_ . cm1	renum	travail . me1
_ . cel		travail . ele
travail.cpl		travail . cpl

-> Traitement de "travail . cpl"

Une opération de simplification des relations est en premier lieu réalisée .

Les couples (< lb > , < Marque >) vont être remplacés par les noeuds adéquats < Noeud > , ceci en utilisant les relations bloc_marquage (< lb > , < Marque > , < Noeud >) .

Les relations bloc_marquage sont ensuite éliminées .

Puis vient l'opération de transfert des relations classiques de type "me1" ou "ele" de "travail . cpl" vers "travail . me1" ou "travail . ele" selon le cas .

La lecture des noeuds permet d'identifier les types de relation < Noeud > -> < ma. > I < mf. >
-> < te. >

Seul les relations inter-domaines restent dans "travail .cpl" .

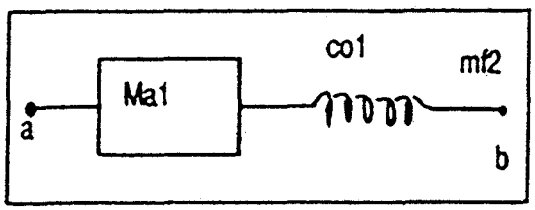
-> Traitement final pour l'obtention du "travail.bgs" à l'issu du couplage

Les blocs "travail.me1" et "travail.ele" sont transformés de façon à obtenir leurs Bond-Graphs sans causalité respectif .

Puis vient l'opération de création du morceau de Bond-Graph issu des relations inter-domaines placées dans "travail . cpl" .

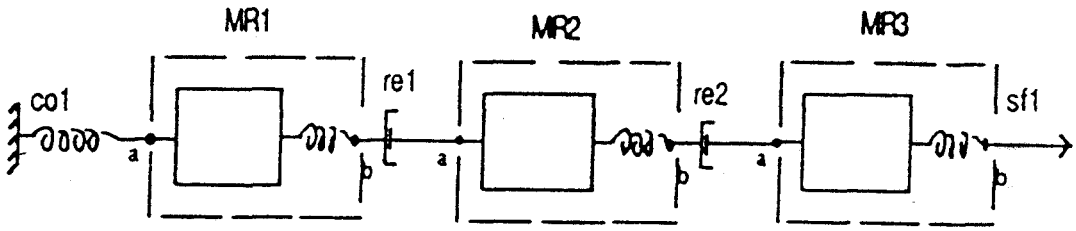
IV) Exemple

Soit un module masse ressort :



modèle ("mr" , "cm1")
rel (ma1 , série (co1) , mf2)
marquage (a , ma1)
marquage (b , mf2)

et considérons le système de couplage suivant



modèle ("scp1" , "cpl")
bloc (mr1 , mr , cm1)
bloc (mr2 , mr , cm1)
bloc (mr3 , mr , cm1)
rel_bb (mr1 , b , série (re1) , mr2 , a)
rel_bb (mr2 , b , série (re2) , mr3 , a)
bati_b (mr1 , a , série (co1))
source_b (mr3 , b , flux (sf1))

Disposant de modèle ("mr" , "cm1") et de modèle ("scp1" , "cpl") , il ne reste plus qu'à réaliser le couplage .

1 ère transformation en balayant les blocs , on a :

① modèle ("travail" , "me1")

rel (ma101 , série (co101) , mf102)
rel (ma201 , série (co201) , mf202)
rel (ma301 , série (co301) , mf302)

```
modèle("travail","cpl")
bloc_marquage ( mr1 , a , ma101 )
bloc_marquage ( mr1 , b , mf102 )
bloc_marquage ( mr2 , a , ma201 )
bloc_marquage ( mr2 , b , mf202 )
bloc_marquage ( mr3 , a , ma301 )
bloc_marquage ( mr3 , b , mf302 )
```

2 ème transformation en servant des blocs_marquage ceci :

```
rel_bb ( mr1 , b , série ( re1 ) , mr2 , a )
rel_bb ( mr2 , b , série ( re2 ) , mr3 , a )
bati_b ( mr1 , a , série ( co1 ) )
source_b ( mr3 , b , flux ( sf1 ) )
```

devient

②
```
rel ( mf102 , série (re1) , ma201 )
rel ( mf202 , série (re2) , ma301 )
bati ( ma101 , série (co1) )
source ( mf302 , flux (sf1) )
```

2 rajouté à 1 donne le modèle utilisateur du système couplé :

modèle ("travail" , "me1")

```
rel ( ma101 , série (co101) , mf102 )
rel ( ma201 , série (co201) , mf202 )
rel ( ma301 , série (co301) , mf302 )
rel ( mf102 , série (re1) , ma201 )
rel ( mf202 , série (re2) , ma301 )
bati ( ma101 , série (co1) )
source ( mf302 , flux (sf1) )
```

Et il ne reste plus qu'à appeler me1bg , simp puis caus , pour obtenir le Bond-Graph simplifié avec causalités !

V) Conclusion

Nous avons réalisé le couplage au niveau modèle utilisateur ,mais cela est aussi possible de le réaliser au niveau Bond-Graph avec une méthodologie similaire .

Les problèmes posés sont différents, concernant en particulier le choix du type de modèle Bond-Graph à traiter, sans simplification,ou aprés simplification, avec ou sans causalité.

Un travail est actuellement en cours sur ce problème. [DEA P. Remy 1990]

D'autres méthodes de couplage restent envisageables tout comme d'autres langages de modèles utilisateurs peuvent être pris en considération . [DEA Ringot 1990]

ANNEXES

Dans cette annexe, nous présentons quatre exemples simples illustrant des constructions complètes allant du modèle utilisateur au modèle sous forme d'équations .

Les données présentées sont donc:

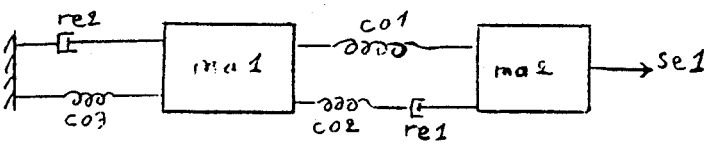
* le système physique et son modèle utilisateur.

* le modèle Bond-Graph non simplifié et simplifié sans et avec causalité.

* le modèle équation sous deux formes (forme classique et forme syntaxe Matlab)

MODELE UTILISATEUR D'UN SYSTEME MECANIQUE DE DIMENSION 1 .
NOM : mec1

```
modele(mec1,mel)
rel(ma1,serie(co1),ma2)
rel(ma1,serie(co2,re1),ma2)
bati(ma1,para(re2,co3))
source(ma2,effort(se1))
```



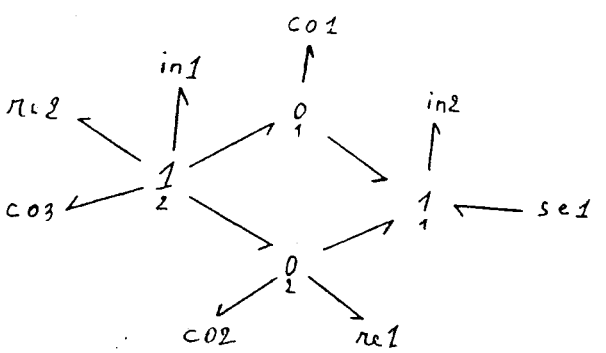
MODELE BOND-GRAPH SANS CAUSALITE . NON SIMPLIFIE
NOM : mec1

```
modele(mec1,bgs)
jonction_1(1,(in2,se1))
jonction_1(2,(in1,re2,co3))
jonction_0(1,(co1))
jonction_0(2,(co2,re1))
lien_01(1,1)
lien_01(2,1)
lien_10(2,1)
lien_10(2,2)
```

Remarque : le noeud de vitesse nulle correspondant au bati est déjà éliminé à cette étape.

MODELE BOND-GRAPH SANS CAUSALITE SIMPLIFIE .
NOM : mec1

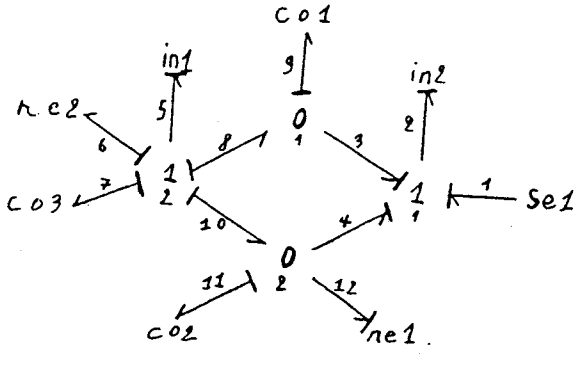
```
modele(mec1,bgp)
jonction_1(1,(in2,se1))
jonction_1(2,(in1,re2,co3))
jonction_0(1,(co1))
jonction_0(2,(co2,re1))
lien_01(1,1)
lien_01(2,1)
lien_10(2,1)
lien_10(2,2)
```



```
-------------------------------------
MODELE BOND-GRAPH AVEC CAUSALITES .
NOM : mec1
-------------------------------------

 modele(mec1,bgc)
 jonction_1(1,(in2,se1))
 jonction_1(2,(in1,re2,co3))
 jonction_0(1,(co1))
 jonction_0(2,(co2,re1))
 lien_01(1,1)
 lien_01(2,1)
 lien_10(2,1)
 lien_10(2,2)
 numero_effort(1,se1,un1)
 numero_effort(2,un1,in2)
 numero_effort(3,zer1,un1)
 numero_effort(4,zer2,un1)
 numero_effort(5,un2,in1)
 numero_effort(6,re2,un2)
 numero_effort(7,co3,un2)
 numero_effort(8,zer1,un2)
 numero_effort(9,co1,zer1)
 numero_effort(10,zer2,un2)
 numero_effort(11,co2,zer2)
 numero_effort(12,zer2,re1)
 nombre_de_liens(12)
```

Le tracé du BG est immédiat.



```
-------------------------
 VECTEURS ET MATRICES .
 NOM : mec1
-------------------------

  modele(mec1,equ)
  dimension_vecteur(Xi',5)
  dimension_vecteur(Din,2)
  dimension_vecteur(sortie_matrice_S,7)
  dimension_vecteur(Zi,5)
  dimension_vecteur(Xd',0)
  dimension_vecteur(Zd,0)
  dimension_vecteur(Dout,2)
  dimension_vecteur(source_U,1)
  dimension_vecteur(entree_matrice_S,8)
  dimensions_matrice(L,2,2)
  dimensions_matrice(Fi,5,5)
  dimensions_matrice(Fd,0,0)
  dimensions_matrice(S,7,8)
  element_vecteur(Xi',1,e2)
  element_vecteur(Xi',2,e5)
  element_vecteur(Xi',3,f7)
  element_vecteur(Xi',4,f9)
  element_vecteur(Xi',5,f11)
  element_vecteur(Din,1,e12)
  element_vecteur(Din,2,f6)
  element_vecteur(sortie_matrice_S,1,e2)
  element_vecteur(sortie_matrice_S,2,e5)
  element_vecteur(sortie_matrice_S,3,f7)
  element_vecteur(sortie_matrice_S,4,f9)
  element_vecteur(sortie_matrice_S,5,f11)
  element_vecteur(sortie_matrice_S,6,e12)
  element_vecteur(sortie_matrice_S,7,f6)
  element_vecteur(Zi,1,f2)
  element_vecteur(Zi,2,f5)
  element_vecteur(Zi,3,e7)
  element_vecteur(Zi,4,e9)
  element_vecteur(Zi,5,e11)
  element_vecteur(Dout,1,f12)
  element_vecteur(Dout,2,e6)
  element_vecteur(source_U,1,e1)
  element_vecteur(entree_matrice_S,1,f2)
  element_vecteur(entree_matrice_S,2,f5)
  element_vecteur(entree_matrice_S,3,e7)
  element_vecteur(entree_matrice_S,4,e9)
  element_vecteur(entree_matrice_S,5,e11)
  element_vecteur(entree_matrice_S,6,f12)
  element_vecteur(entree_matrice_S,7,e6)
  element_vecteur(entree_matrice_S,8,e1)
  element_matrice(L,1,1,1/re1)
  element_matrice(L,2,2,re2)
  element_matrice(Fi,1,1,1/in2)
  element_matrice(Fi,2,2,1/in1)
  element_matrice(Fi,3,3,1/co3)
  element_matrice(Fi,4,4,1/co1)
  element_matrice(Fi,5,5,1/co2)
  element_matrice(S,1,4,+1)
  element_matrice(S,1,5,+1)
  element_matrice(S,1,8,+1)
  element_matrice(S,2,7,-1)
```

```
element_matrice(S,2,3,-1)
element_matrice(S,2,4,-1)
element_matrice(S,2,5,-1)
element_matrice(S,3,2,+1)
element_matrice(S,4,2,+1)
element_matrice(S,4,1,-1)
element_matrice(S,5,6,-1)
element_matrice(S,5,1,-1)
element_matrice(S,5,2,+1)
element_matrice(S,6,5,+1)
element_matrice(S,7,2,+1)
```

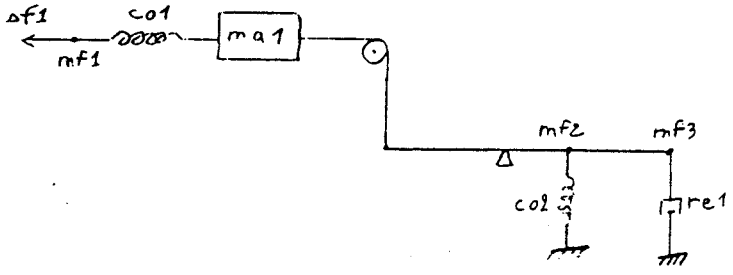
```
------------------------
 VECTEURS ET MATRICES .
 NOM : mec1
------------------------

ecteur Xi' :
 e2 ; e5 ; f7 ; f9 ; f11 ]
ecteur Din :
 e12 ; f6 ]
ecteur de sortie de la matrice S :
 e2 ; e5 ; f7 ; f9 ; f11 ; e12 ; f6 ]
cteur Zi :
 f2 ; f5 ; e7 ; e9 ; e11 ]
cteur Zd :
 ]
cteur Xd' :
 ]
cteur Dout :
 f12 ; e6 ]
cteur source U :
 e1 ]
cteur d'entrée de la matrice S :
 f2 ; f5 ; e7 ; e9 ; e11 ; f12 ; e6 ; e1 ]
trice L telle que Dout = L Din :
 1/re1 0 ;
 0 re2 ]
trice Fi telle que Zi = Fi Xi :
 1/in2 0 0 0 0 ;
 0 1/in1 0 0 0 ;
 0 0 1/co3 0 0 ;
 0 0 0 1/co1 0 ;
 0 0 0 0 1/co2 ]
trice Fd telle que Zd = Fd Xd :
 ]
trice S :
 0 0 0 +1 +1 0 0 +1 ;
 0 0 -1 -1 -1 0 -1 0 ;
 0 +1 0 0 0 0 0 0 ;
 -1 +1 0 0 0 0 0 0 ;
 -1 +1 0 0 0 -1 0 0 ;
 0 0 0 0 +1 0 0 0 ;
 0 +1 0 0 0 0 0 0 ]

-------------------------------------------------------------
```

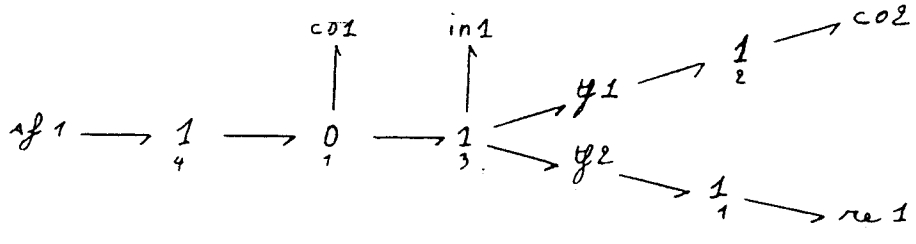
MODELE UTILISATEUR D'UN SYSTEME MECANIQUE DE DIMENSION 1 .
NOM : mec2

```
modele(mec2,me1)
rel(mf1,serie(co1),ma1)
rel(ma1,transfo(tf1),mf2)
rel(ma1,transfo(tf2),mf3)
bati(mf2,serie(co2))
bati(mf3,serie(re1))
source(mf1,flux(sf1))
```



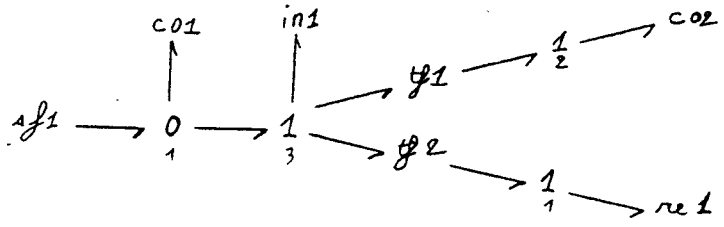
MODELE BOND-GRAPH SANS CAUSALITE . NON SIMPLIFIE
NOM : mec2

```
modele(mec2,bgs)
jonction_1(1,(re1))
jonction_1(2,(co2))
jonction_1(3,(in1))
jonction_1(4,(sf1))
jonction_0(1,(co1))
jonction_tf(1)
jonction_tf(2)
lien_01(1,3)
lien_10(4,1)
lien_1tf(3,1)
lien_1tf(3,2)
lien_tf1(1,2)
lien_tf1(2,1)
```



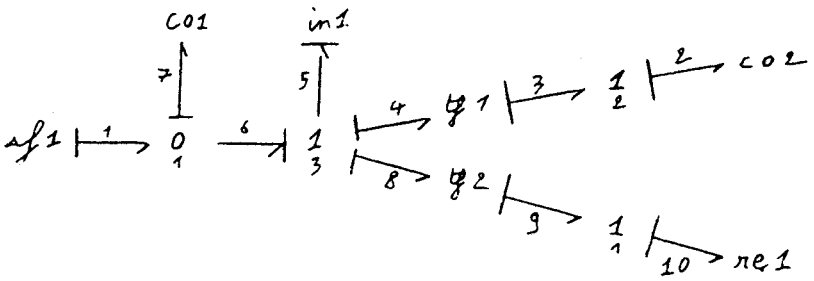
MODELE BOND-GRAPH SANS CAUSALITE SIMPLIFIE .
NOM : mec2

```
modele(mec2,bgp)
jonction_1(1,(re1))
jonction_1(2,(co2))
jonction_1(3,(in1))
jonction_0(1,(sf1,co1))
jonction_tf(1)
jonction_tf(2)
lien_01(1,3)
lien_1tf(3,1)
lien_1tf(3,2)
lien_tf1(1,2)
lien_tf1(2,1)
```



MODELE BOND-GRAPH AVEC CAUSALITES .
NOM : mec2

```
modele(mec2,bgc)
jonction_1(1,(re1))
jonction_1(2,(co2))
jonction_1(3,(in1))
jonction_0(1,(sf1,co1))
jonction_tf(1)
jonction_tf(2)
lien_01(1,3)
lien_1tf(3,1)
lien_1tf(3,2)
lien_tf1(1,2)
lien_tf1(2,1)
numero_effort(1,zer1,sf1)
numero_effort(2,co2,un2)
numero_effort(3,un2,tf1)
numero_effort(4,tf1,un3)
numero_effort(5,un3,in1)
numero_effort(6,zer1,un3)
numero_effort(7,co1,zer1)
numero_effort(8,tf2,un3)
numero_effort(9,un1,tf2)
numero_effort(10,re1,un1)
nombre_de_liens(10)
```



```
--------------------------
 VECTEURS ET MATRICES .
 NOM : mec2
--------------------------

  modele(mec2,equ)
  dimension_vecteur(Xi',3)
  dimension_vecteur(Din,1)
  dimension_vecteur(sortie_matrice_S,4)
  dimension_vecteur(Zi,3)
  dimension_vecteur(Xd',0)
  dimension_vecteur(Zd,0)
  dimension_vecteur(Dout,1)
  dimension_vecteur(source_U,1)
  dimension_vecteur(entree_matrice_S,5)
  dimensions_matrice(L,1,1)
  dimensions_matrice(Fi,3,3)
  dimensions_matrice(Fd,0,0)
  dimensions_matrice(S,4,5)
  element_vecteur(Xi',1,e5)
  element_vecteur(Xi',2,f2)
  element_vecteur(Xi',3,f7)
  element_vecteur(Din,1,f10)
  element_vecteur(sortie_matrice_S,1,e5)
  element_vecteur(sortie_matrice_S,2,f2)
  element_vecteur(sortie_matrice_S,3,f7)
  element_vecteur(sortie_matrice_S,4,f10)
  element_vecteur(Zi,1,f5)
  element_vecteur(Zi,2,e2)
  element_vecteur(Zi,3,e7)
  element_vecteur(Dout,1,e10)
  element_vecteur(source_U,1,f1)
  element_vecteur(entree_matrice_S,1,f5)
  element_vecteur(entree_matrice_S,2,e2)
  element_vecteur(entree_matrice_S,3,e7)
  element_vecteur(entree_matrice_S,4,e10)
  element_vecteur(entree_matrice_S,5,f1)
  element_matrice(L,1,1,re1)
  element_matrice(Fi,1,1,1/in1)
  element_matrice(Fi,2,2,1/co2)
  element_matrice(Fi,3,3,1/co1)
  element_matrice(S,1,3,+1)
  element_matrice(S,1,2,-1/tf1)
  element_matrice(S,1,4,-1/tf2)
  element_matrice(S,2,1,+1/tf1)
  element_matrice(S,3,5,+1)
  element_matrice(S,3,1,-1)
  element_matrice(S,4,1,+1/tf2)

-------------------------------------------------------------
```

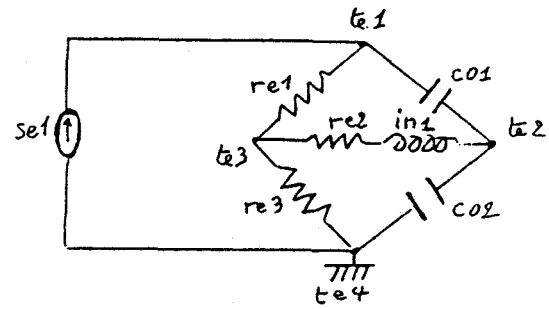
```
-------------------------
 VECTEURS ET MATRICES .
 NOM : mec2
-------------------------

Vecteur Xi' :
[ e5 ; f2 ; f7 ]
Vecteur Din :
[ f10 ]
Vecteur de sortie de la matrice S :
[ e5 ; f2 ; f7 ; f10 ]
Vecteur Zi :
[ f5 ; e2 ; e7 ]
Vecteur Zd :
[ ]
Vecteur Xd' :
[ ]
Vecteur Dout :
[ e10 ]
Vecteur source U :
[ f1 ]
Vecteur d'entrée de la matrice S :
[ f5 ; e2 ; e7 ; e10 ; f1 ]
Matrice L telle que Dout = L Din :
[ re1 ]
Matrice Fi telle que Zi = Fi Xi :
[ 1/in1 0 0 ;
  0 1/co2 0 ;
  0 0 1/co1 ]
Matrice Fd telle que Zd = Fd Xd :
[ ]
Matrice S :
[ 0 -1/tf1 +1 -1/tf2 0 ;
  +1/tf1 0 0 0 0 ;
  -1 0 0 0 +1 ;
  +1/tf2 0 0 0 0 ]

-------------------------------------------------------------
```

MODELE UTILISATEUR D'UN SYSTEME ELECTRIQUE .
NOM : elec1

```
modele(elec1,ele)
rel(te1,serie(co1),te2)
rel(te1,serie(re1),te3)
rel(te3,serie(re2,in1),te2)
rel(te3,serie(re3),te4)
rel(te2,serie(co2),te4)
rel(te4,serie(se1),te1)
point_de_masse(te4)
```

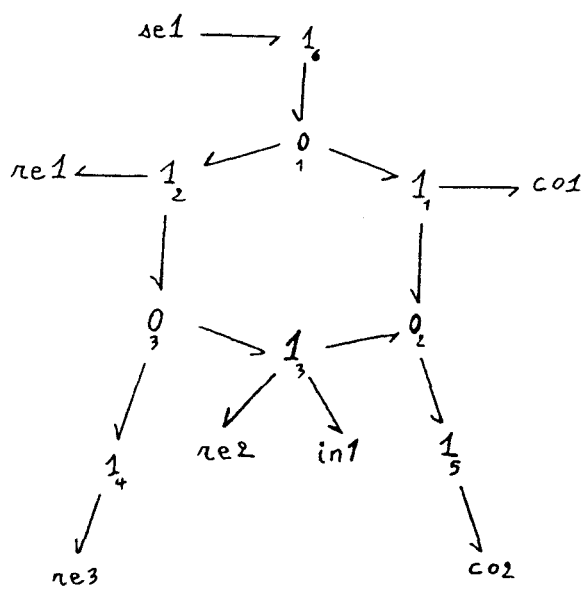


MODELE BOND-GRAPH SANS CAUSALITE .
NOM : elec1

NON SIMPLIFIE

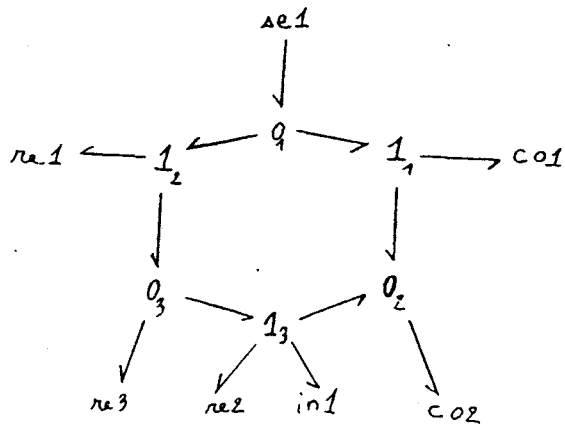
```
modele(elec1,bgs)
jonction_1(1,(co1))
jonction_1(2,(re1))
jonction_1(3,(re2,in1))
jonction_1(4,(re3))
jonction_1(5,(co2))
jonction_1(6,(se1))
jonction_0(2,())
jonction_0(3,())
jonction_0(1,())
lien_01(1,1)
lien_01(1,2)
lien_01(3,3)
lien_01(3,4)
lien_01(2,5)
lien_10(1,2)
lien_10(2,3)
lien_10(3,2)
lien_10(6,1)
```

La non simplification ici suppose déjà l'élimination du point de masse.



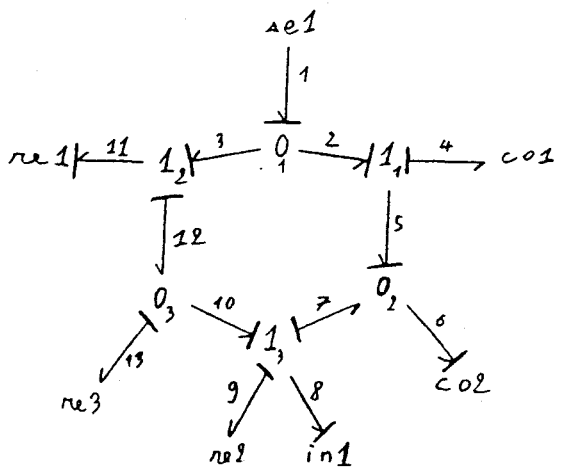
MODELE BOND-GRAPH SANS CAUSALITE SIMPLIFIE .
NOM : elec1

```
modele(elec1,bgp)
jonction_1(1,(co1))
jonction_1(2,(re1))
jonction_1(3,(re2,in1))
jonction_0(3,(re3))
jonction_0(2,(co2))
jonction_0(1,(se1))
lien_01(1,1)
lien_01(1,2)
lien_01(3,3)
lien_10(1,2)
lien_10(2,3)
lien_10(3,2)
```



MODELE BOND-GRAPH AVEC CAUSALITES .
NOM : elec1

```
modele(elec1,bgc)
jonction_1(1,(co1))
jonction_1(2,(re1))
jonction_1(3,(re2,in1))
jonction_0(3,(re3))
jonction_0(2,(co2))
jonction_0(1,(se1))
lien_01(1,1)
lien_01(1,2)
lien_01(3,3)
lien_10(1,2)
lien_10(2,3)
lien_10(3,2)
causalite_derivee(co2)
numero_effort(1,se1,zer1)
numero_effort(2,zer1,un1)
numero_effort(3,zer1,un2)
numero_effort(4,co1,un1)
numero_effort(5,un1,zer2)
numero_effort(6,zer2,co2)
numero_effort(7,zer2,un3)
numero_effort(8,un3,in1)
numero_effort(9,re2,un3)
numero_effort(10,zer3,un3)
numero_effort(11,un2,re1)
numero_effort(12,zer3,un2)
numero_effort(13,re3,zer3)
nombre_de_liens(13)
```



```
-------------------------
 VECTEURS ET MATRICES .
 NOM : elec1
-------------------------

  modele(elec1,equ)
  dimension_vecteur(Xi',2)
  dimension_vecteur(Din,3)
  dimension_vecteur(sortie_matrice_S,5)
  dimension_vecteur(Zi,2)
  dimension_vecteur(Xd',1)
  dimension_vecteur(Zd,1)
  dimension_vecteur(Dout,3)
  dimension_vecteur(source_U,1)
  dimension_vecteur(entree_matrice_S,7)
  dimensions_matrice(L,3,3)
  dimensions_matrice(Fi,2,2)
  dimensions_matrice(Fd,1,1)
  dimensions_matrice(S,5,7)
  element_vecteur(Xi',1,e8)
  element_vecteur(Xi',2,f4)
  element_vecteur(Din,1,e11)
  element_vecteur(Din,2,f9)
  element_vecteur(Din,3,f13)
  element_vecteur(sortie_matrice_S,1,e8)
  element_vecteur(sortie_matrice_S,2,f4)
  element_vecteur(sortie_matrice_S,3,e11)
  element_vecteur(sortie_matrice_S,4,f9)
  element_vecteur(sortie_matrice_S,5,f13)
  element_vecteur(Zi,1,f8)
  element_vecteur(Zi,2,e4)
  element_vecteur(Xd',1,f6)
  element_vecteur(Zd,1,e6)
  element_vecteur(Dout,1,f11)
  element_vecteur(Dout,2,e9)
  element_vecteur(Dout,3,e13)
  element_vecteur(source_U,1,e1)
  element_vecteur(entree_matrice_S,1,f8)
  element_vecteur(entree_matrice_S,2,e4)
  element_vecteur(entree_matrice_S,3,f6)
  element_vecteur(entree_matrice_S,4,f11)
  element_vecteur(entree_matrice_S,5,e9)
  element_vecteur(entree_matrice_S,6,e13)
  element_vecteur(entree_matrice_S,7,e1)
  element_matrice(L,1,1,1/re1)
  element_matrice(L,2,2,re2)
  element_matrice(L,3,3,re3)
  element_matrice(Fi,1,1,1/in1)
  element_matrice(Fi,2,2,1/co1)
  element_matrice(Fd,1,1,1/co2)
  element_matrice(S,1,5,-1)
  element_matrice(S,1,6,+1)
  element_matrice(S,1,2,+1)
  element_matrice(S,1,7,-1)
  element_matrice(S,2,3,+1)
  element_matrice(S,2,1,-1)
  element_matrice(S,3,6,-1)
  element_matrice(S,3,7,+1)
  element_matrice(S,4,1,+1)
  element_matrice(S,5,1,-1)
  element_matrice(S,5,4,+1)
```

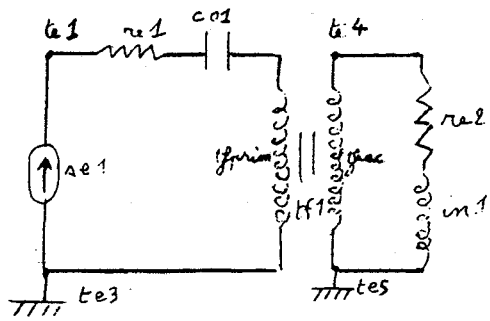
```
-------------------------
 VECTEURS ET MATRICES .
 NOM : elec1
-------------------------

Vecteur Xi' :
[ e8 ; f4 ]
Vecteur Din :
[ e11 ; f9 ; f13 ]
Vecteur de sortie de la matrice S :
[ e8 ; f4 ; e11 ; f9 ; f13 ]
Vecteur Zi :
[ f8 ; e4 ]
Vecteur Zd :
[ e6 ]
Vecteur Xd' :
[ f6 ]
Vecteur Dout :
[ f11 ; e9 ; e13 ]
Vecteur source U :
[ e1 ]
Vecteur d'entrée de la matrice S :
[ f8 ; e4 ; f6 ; f11 ; e9 ; e13 ; e1 ]
Matrice L telle que Dout = L Din :
[ 1/re1 0 0 ;
  0 re2 0 ;
  0 0 re3 ]
Matrice Fi telle que Zi = Fi Xi :
[ 1/in1 0 ;
  0 1/co1 ]
Matrice Fd telle que Zd = Fd Xd :
[ 1/co2 ]
Matrice S :
[ 0 +1 0 0 -1 +1 -1 ;
  -1 0 +1 0 0 0 0 ;
  0 0 0 0 0 -1 +1 ;
  +1 0 0 0 0 0 0 ;
  -1 0 0 +1 0 0 0 ]

-------------------------------------------------------------
```

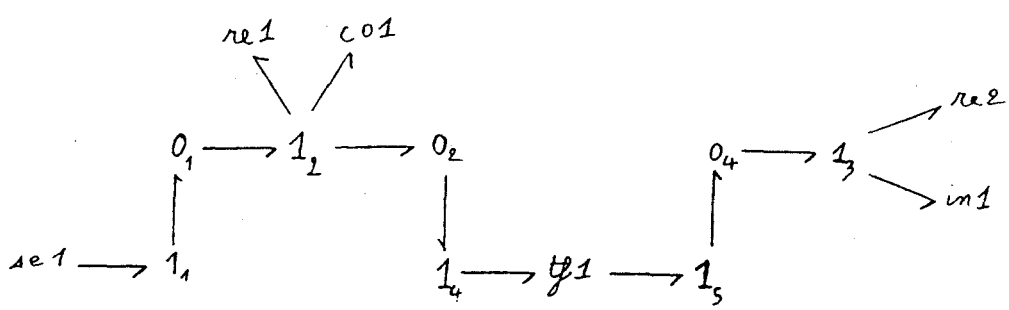
MODELE UTILISATEUR D'UN SYSTEME ELECTRIQUE .
NOM : elec4

```
modele(elec4,ele)
rel(te3,serie(se1),te1)
rel(te1,serie(re1,co1),te2)
rel(te2,tfprim(tf1),te3)
rel(te5,tfsec(tf1),te4)
rel(te4,serie(re2,in1),te5)
point_de_masse(te3)
point_de_masse(te5)
```



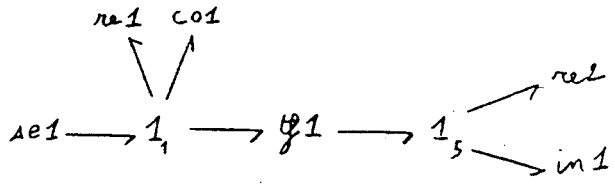
MODELE BOND-GRAPH SANS CAUSALITE . NON SIMPLIFIE
NOM : elec4

```
modele(elec4,bgs)
jonction_1(1,(se1))
jonction_1(2,(re1,co1))
jonction_1(3,(re2,in1))
jonction_1(4,())
jonction_1(5,())
jonction_0(4,())
jonction_0(2,())
jonction_0(1,())
jonction_tf(1)
lien_01(1,2)
lien_01(4,3)
lien_01(2,4)
lien_10(1,1)
lien_10(2,2)
lien_10(5,4)
lien_1tf(4,1)
lien_tf1(1,5)
```



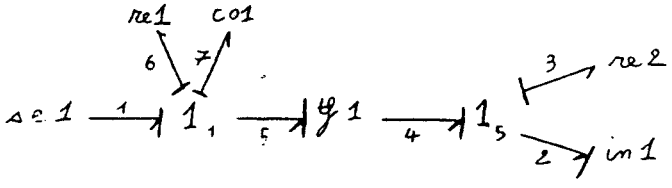
MODELE BOND-GRAPH SANS CAUSALITE SIMPLIFIE .
NOM : elec4

```
modele(elec4,bgp)
jonction_1(5,(re2,in1))
jonction_1(1,(se1,re1,co1))
jonction_tf(1)
lien_1tf(1,1)
lien_tf1(1,5)
```



MODELE BOND-GRAPH AVEC CAUSALITES .
NOM : elec4

```
modele(elec4,bgc)
jonction_1(5,(re2,in1))
jonction_1(1,(se1,re1,co1))
jonction_tf(1)
lien_1tf(1,1)
lien_tf1(1,5)
numero_effort(1,se1,un1)
numero_effort(2,un5,in1)
numero_effort(3,re2,un5)
numero_effort(4,tf1,un5)
numero_effort(5,un1,tf1)
numero_effort(6,re1,un1)
numero_effort(7,co1,un1)
nombre_de_liens(7)
```



```
--------------------------
VECTEURS ET MATRICES .
NOM : elec4
--------------------------

  modele(elec4,equ)
  dimension_vecteur(Xi',2)
  dimension_vecteur(Din,2)
  dimension_vecteur(sortie_matrice_S,4)
  dimension_vecteur(Zi,2)
  dimension_vecteur(Xd',0)
  dimension_vecteur(Zd,0)
  dimension_vecteur(Dout,2)
  dimension_vecteur(source_U,1)
  dimension_vecteur(entree_matrice_S,5)
  dimensions_matrice(L,2,2)
  dimensions_matrice(Fi,2,2)
  dimensions_matrice(Fd,0,0)
  dimensions_matrice(S,4,5)
  element_vecteur(Xi',1,e2)
  element_vecteur(Xi',2,f7)
  element_vecteur(Din,1,f3)
  element_vecteur(Din,2,f6)
  element_vecteur(sortie_matrice_S,1,e2)
  element_vecteur(sortie_matrice_S,2,f7)
  element_vecteur(sortie_matrice_S,3,f3)
  element_vecteur(sortie_matrice_S,4,f6)
  element_vecteur(Zi,1,f2)
  element_vecteur(Zi,2,e7)
  element_vecteur(Dout,1,e3)
  element_vecteur(Dout,2,e6)
  element_vecteur(source_U,1,e1)
  element_vecteur(entree_matrice_S,1,f2)
  element_vecteur(entree_matrice_S,2,e7)
  element_vecteur(entree_matrice_S,3,e3)
  element_vecteur(entree_matrice_S,4,e6)
  element_vecteur(entree_matrice_S,5,e1)
  element_matrice(L,1,1,re2)
  element_matrice(L,2,2,re1)
  element_matrice(Fi,1,1,1/in1)
  element_matrice(Fi,2,2,1/co1)
  element_matrice(S,1,5,+tf1)
  element_matrice(S,1,4,-tf1)
  element_matrice(S,1,2,-tf1)
  element_matrice(S,1,3,-1)
  element_matrice(S,2,1,+tf1)
  element_matrice(S,3,1,+1)
  element_matrice(S,4,1,+tf1)

-------------------------------------------------------------
```

```
-------------------------
 VECTEURS ET MATRICES .
 NOM : elec4
-------------------------

Vecteur Xi' :
[ e2 ; f7 ]
Vecteur Din :
[ f3 ; f6 ]
Vecteur de sortie de la matrice S :
[ e2 ; f7 ; f3 ; f6 ]
Vecteur Zi :
[ f2 ; e7 ]
Vecteur Zd :
[ ]
Vecteur Xd' :
[ ]
Vecteur Dout :
[ e3 ; e6 ]
Vecteur source U :
[ e1 ]
Vecteur d'entrée de la matrice S :
[ f2 ; e7 ; e3 ; e6 ; e1 ]
Matrice L telle que Dout = L Din :
[ re2 0 ;
  0 re1 ]
Matrice Fi telle que Zi = Fi Xi :
[ 1/in1 0 ;
  0 1/co1 ]
Matrice Fd telle que Zd = Fd Xd :
[ ]
Matrice S :
[ 0 -tf1 -1 -tf1 +tf1 ;
  +tf1 0 0 0 0 ;
  +1 0 0 0 0 ;
  +tf1 0 0 0 0 ]
```

CONCLUSION GENERALE

Conclusion générale

Avec le développement d'Archer la modélisation des systèmes dynamiques par l'approche Bond-Graph emboite le pas des systèmes experts.

La maquette construite n'est pas terminée; nous envisageons son avenir avec confiance de part le fait que ses qualités modulaires la rendent facilement extensible.

Le couplage entre modèles de base de données reste un problème délicat à traiter, ce problème est en cours d'étude.

De plus l'interfaçage avec d'autres logiciels sera à réaliser, en particulier avec des logiciels de simulation.

Pour ce qui concerne l'interface utilisateur l'utilisation de modules graphiques pour l'entrée de schémas physiques est en cours de développement.

Le choix de Turbo-Prolog sur compatible Pc rend ARCHER trés portable et d'un accés trés aisé.

Sa construction modulaire permet,à chaque instant,de n'occuper qu'une partie de la mémoire vive,et le rend exportable sans probleme sous DOS, qui gère au maximum 640 KO de mémoire RAM.

La taille de chacun des modules est approximativement de 70 KO,et pour le module d'édition de 200 KO.

L'introduction de procédures d'analyse développées par [SUEUR C. 1990] renforcera son caractère d'expertise et sa spécificité parmi les logiciels de modélisation existants.

BIBLIOGRAPHIE .

AZMANI A , BOUAYAD R , DAUPHIN TANGUY G(1990)
Artificial intelligence approach for the causal analysis
of Bond-Graph models
IMACS IFAC MIM-S21-90 Symposium,Bruxelles ,sept 1990
pp V-B-3-1,V-B-3-6

BEUKEBOOM JJ , VAN DIXHOORN JJ ,MEERMAN Jw (1985)
Simulation of mixed Bond-Graphs and block diagrams
on personal computer using Tutsim
Journal of the Franklin Institute , vol 319
n°1/2,pp 257 - 267

BOUAYAD R , AZMANI A , DAUPHIN TANGUY G (1990)
Algorithm for the computer aided drawing of Bond-Graph
models
IMACS IFAC MIM-S21-90 Symposium,Bruxelles ,sept 1990
pp II-B-4-1,II-B-4-5

BREEDVELD Pc (1979)
Irreversible thermodynamics and Bond-Graph,
M . Sc thesis university of Enschede, the Netherlands

BRESCH D (1986)
Etude et réalisation d'un outil logiciel de modelisation et de
simulation de systèmes physiques utilisant la représentation
des graphes à liens .
Thèse de docteur en électronique,universite de Haute Alsace /
Juin 1986 .

BROENINK J F (1986)
A Bond-Graph based modelling language
Complex and distributed systems :analysis and control,vol 4
IMACS Transactions on scientific computation
Tzafestas-Borne editeurs,
pp 81-86 North Holand,amsterdam

BROENINK Jan F (1986)
Sidops,a bond graph based modelling language
complex and distributed system : analysis ,
simulation and control.
Elsevier Science Publissher B.V. (North - Holland)
(c) Imacs , 1986 .

BROENINK Jan F
Outline of Camas : a computer aided Modelling
Analysis and Simulation Environment .
2 d European Simulation Congress
Antwerpen Sept 1986 .

COUTEREEL L, DAUPHIN - TANGUY G, SUEUR C
Presentation of a processor for the computer
aided Modelling of dynamical systems
through a Bond-Graph approach.
Imacs internationnal Symposium on AI expert system
and langage in Modelling and Simulation .
Barcelona 2-4 june1987 ,pp.197_202

COUTEREEL L, DAUPHIN - TANGUY G, BOUAYAD R
Artificial Intelligence and Bond Graph Methodology
for the modelling of dynamical systems .
12 th Imacs World Congress , Paris 1988 , pp.35-37 .

DELAHAYE J P (1987)
Systèmes experts : organisation et programmation
des bases de connaissances en calcul propositionnel.
Edition Eyrolles , (1987) .

FARRENY H
Les systèmes experts , principes et exemples .
Cepadues - editions , (1985) .

FARRENY H, GHALLAB M
Eléments d'intelligence artificielle
Traité de nouvelles technologies,
Série intelligence artificielle.
Edition Hermes , (1987) .

GRANDA JJ (1985)
Computer generation of physical System
differential equations using Bond-Graphs.
Journal of the Franklin Institute,vol 319,
n°1/2 pp 243--255

KARNOPP D C, ROSENBERG R C
System dynamics : a unified approach
John Wiley and Sons , New York (1975) .

PAYNTER H.M (1961)
Analysis and design of engineering system.
MIT Press , Cambridge , Massachussets

REMY P. (1990)
Elaboration d'un module de couplage entre Bond-Graph pour le processeur ARCHER.
Dea de productique, université Lille I.

RINGOT D. (1990)
Analyse syntaxique et sémantique d'un langage utilisateur pour la construction des Bond-Graphs.
Dea de productique, université Lille I.

ROSENBERG R C, KARNOPP D C (1983)
Introduction to Physical System Dynamics .
Mc Gray-Hill book Company - Series in Mechanical Engineering . (1983) .

SUEUR C. (1990)
Contribution à la modélisation et à l'analyse des systèmes dynamiques par une approche Bond-Graph.
Application aux systèmes polyarticulés plans à segments flexibles.
Thèse de doctorat d'université, Lille I,n° 584
Octobre 1990 .

THOMA j U (1975)
Introduction to Bond-Graphs and their applications .
Pergamon Press , New York (1975) .

Résumé

Les travaux présentés dans ce mémoire contribuent au développement des systèmes experts liés à la Modélisation de systèmes physiques dynamiques.

Ce processeur ARCHER utilise conjointement les techniques de l'intelligence Artificielle et l'outil Bond-Graph.

ARCHER permet, à partir d'une description en langage utilisateur proche du langage naturel du système physique, la construction du modèle Bond-Graph et la détermination de l'équation d'état associée sous forme d'expressions formelles.

Ecrit en Turbo-Prolog et développé sur Pc, ARCHER possède, par sa construction même, certaines qualités d'un système expert spécialisé.

Les domaines physiques concernés par le processeur sont les systèmes mécaniques en dimension 1 et les systèmes électriques linéaires.

Mots-clefs

- Bond-Graphs
- Système Dynamique
- Modélisation
- Système Expert