

Approche Distribuée et Réactive de la Conception des Systèmes d'Assemblage

THÈSE

présentée à la Faculté des Sciences, pour obtenir
le grade de Docteur ès Sciences, par

François Sprumont

UNIVERSITÉ DE NEUCHÂTEL
Institut d'Informatique et
Intelligence Artificielle
rue Émile-Argand 11
2007 Neuchâtel
Suisse

le 16 mai 1997

Membres du jury:

Jean-Pierre Müller	Professeur à l'IIIA (directeur de thèse)
Alain Bourjault	Directeur et Professeur au Laboratoire d'Automatique de Besançon
Jacques Jacot	Professeur à l'IMT de l'École Polytech- nique Fédérale de Lausanne
Hans-Heinrich Nägeli	Professeur à l'IIIA
Jean-Claude Prélaz	Directeur de Sysmelec SA

IMPRIMATUR POUR LA THÈSE

Approche distribuée et réactive de la conception des
systèmes d'assemblage

de M. François Sprumont

UNIVERSITÉ DE NEUCHÂTEL
FACULTÉ DES SCIENCES

La Faculté des sciences de l'Université de
Neuchâtel sur le rapport des membres du jury,

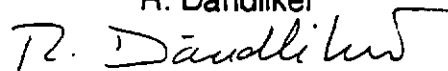
MM. J.-P. Müller (directeur de thèse),
H.-H. Naegeli, J.-C. Prélaz (Sysmelec SA),
A. Bourjault (Besançon) et J. Jacot (EPFL)

autorise l'impression de la présente thèse.

Neuchâtel, le 21 mai 1997

Le doyen:

R. Dändliker



Il n'est pas préférable pour les
hommes de connaître plus qu'ils
ne doivent

Héraclite (110)

L'important n'est pas de connaître
la réponse, mais de comprendre la
question

Lao-Tsen

Avant Propos

Notations

- Une majuscule désigne un ensemble: par exemples, A désigne l'ensemble des arcs d'un graphe, S , l'ensemble de ses sommets, etc.
- Une minuscule désigne un élément ou un objet: par exemples, a désigne un arc d'un graphe, i , un flot d'assemblage, etc.
- Un indice est un qualificatif pour l'objet désigné: par exemples, A_p désigne l'ensemble des agents-postes, i_{crt} désigne l'flot d'assemblage courant, etc.
- Un exposant est un identificateur pour l'objet désigné, soit en tant qu'individu, soit en tant qu'élément d'un autre individu: par exemples, a_i^k désigne l'agent-tâche k , C^{a_i} , désigne l'ensemble des connaissances de l'agent-flot a_i , etc.
- Un graphe est désigné par la police de caractère "courrier": par exemple, G_p désigne un graphe de précedence.

Exemple de la valve de vélo

Afin d'illustrer les notions présentées, nous utilisons l'exemple d'une valve de vélo dont le dessin d'ensemble est donné à la figure 0.1.

Cette valve est composée de cinq pièces:

- *Un corps* (abrégié "Co")
- *Un guide* (abrégié "Gu"): le guide est chassé dans le corps. Sa face supérieure est percée d'un jour rectangulaire dans lequel viennent s'introduire les faces planes de la soupape.
- *Une soupape* (abrégié "So"): la soupape est simplement introduite dans le corps avec lequel elle n'a pas de contact direct. La soupape est bloquée dans le corps par le guide ou par l'écrou.
- *Un joint* (abrégié "Jo"): le joint est chassé sur la soupape. Cette opération n'est plus possible une fois la soupape introduite dans le corps.
- *Un écrou* (abrégié "Ec"): l'écrou est vissé sur la soupape. Il a pour fonction de bloquer la soupape en la tirant vers le haut, ce qui écrase le joint contre le corps et empêche l'air de passer.

A l'assemblage de ces cinq pièces, s'ajoute une opération de sertissage du filetage de la soupape, une fois l'écrou vissé: le sommet du filetage est déformé mécaniquement sur quelques millimètres, afin que l'écrou ne puisse plus être enlevé.

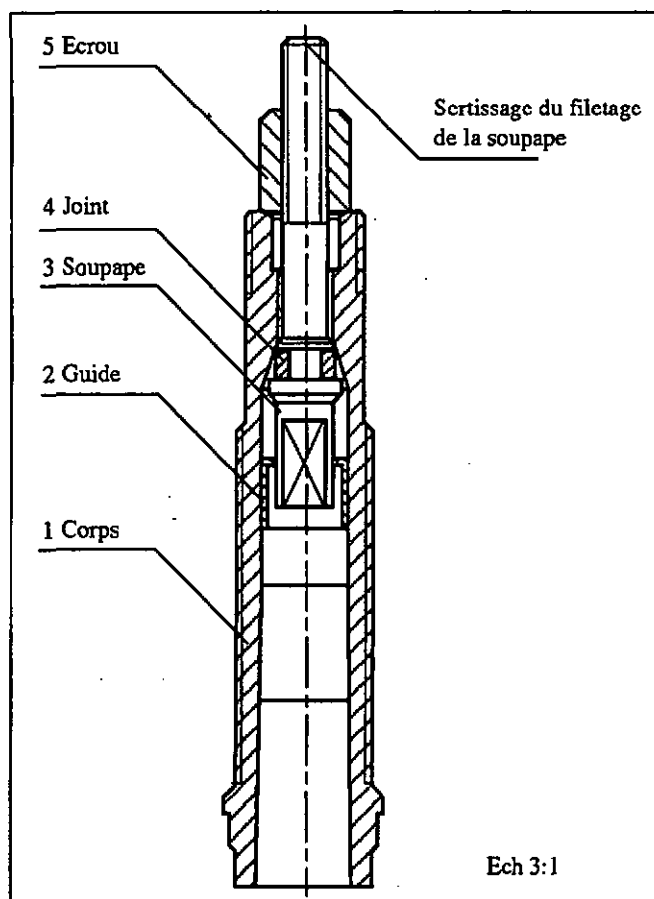


Figure 0.1: Dessin d'ensemble de la valve de vélo

Remerciements

Les travaux présentés dans ce mémoire ont été effectués dans le cadre du projet CERS no 2076.1 à l'Institut d'Informatique et Intelligence Artificielle de l'Université de Neuchâtel. Ils ont été menés en collaboration avec Sysmelec SA, Neuchâtel, un constructeur de systèmes d'assemblage.

Je tiens à remercier ici toutes les personnes sans qui, directement ou indirectement, ce travail n'aurait peut-être pas pu se concrétiser sous cette forme:

- M. Jean-Pierre Müller, professeur à l'IIIA et directeur de cette thèse, pour la confiance qu'il m'a faite, et pour ses analyses éclairées lorsque tout me paraissait sombre.

- M. Jean-Claude Prélaz, directeur de Sysmelec SA, pour avoir permis la collaboration, pour ses conseils de praticien, et pour avoir jugé ce travail en tant que membre du jury.
- M. Patrick von Meiss, de Sysmelec SA, pour avoir passé des heures à maltraiter le prototype d'AMACOLA, et pour ses conseils pour l'interface.
- M. Alain Bourjault, professeur et directeur du Laboratoire d'Automatique de Besançon, pour m'avoir permis d'effectuer un stage profitable au LAB, et pour avoir accepté d'être membre du jury.
- M. Christophe Perrard, du LAB, et à travers lui tous les membres du LAB, pour l'encadrement qu'ils m'ont offert au cours de mon stage.
- M. Hans Heinrich Nägeli, professeur à l'IIIA, en tant que membre du jury.
- M. Jacques Jacot, professeur à l'Institut de Microtechnique de l'EPFL, pour ses conseils et pour avoir jugé ce travail.
- M. Daniel Rufflé, étudiant à l'IIIA, pour avoir développé la plupart des fonctions de visualisation des pièces et de calcul des tolérances d'AMACOLA, au cours d'un projet de semestre.
- Tous mes collègues de l'IIIA, dont les discussions ou les silences, les coups de gueule et les fous-rires, ont constitué le terreau propice à cette thèse: Miguel, Yoël, Miriam, Bernard(s), Claude, Thouraya, Antoine, Nourredine(s), Pontien, Hervé, Saïd, François, Sylvain, Robin, Yassine, Laurent, Gianfranca, ... sans oublier le professeur Pierre-Jean Erard.
- Sandra, enfin, qui malgré les hauts et les bas, a toujours su trouver les attentions qui convenaient à ma vie de thésard un peu décousue, et pour avoir osé me comprendre.

Je tiens également à remercier mes parents, sans qui, forcément, rien n'aurait été possible, et je leur dédie cette thèse, en espérant qu'elle puisse modestement contribuer à l'apaisement des soucis que j'ai pu leur causer.

Table des matières

1	Introduction	13
I	Etat de l'art	15
2	L'approche multi-agents	17
2.1	Introduction	17
2.2	Définitions	18
2.3	Les agents cognitifs	19
2.4	Les agents réactifs	19
2.5	Les modes de communication	20
2.6	L'éco-résolution	20
2.7	Conclusion	22
3	Les problèmes combinatoires	23
3.1	Introduction	23
3.2	Définitions	24
3.3	L'espace de recherche	25
3.4	Le parcours de l'espace	26
3.5	Problèmes combinatoires et SMA	27
3.6	Conclusion	28
4	La conception	29
4.1	Introduction	29
4.2	Typologie des problèmes de conception	30
4.3	Définitions	30
4.4	Le processus de conception	32
4.4.1	L'espace fonctionnel	33
4.4.2	L'espace physique	33
4.4.3	Les liens entre espaces physique et fonctionnel	34
4.5	Les méthodes de la conception courante	34
4.5.1	Les méthodes non basées connaissance	35
4.5.2	Les méthodes basées connaissances	35
4.6	Les méthodes en conception innovatrice	36
4.6.1	Le raisonnement par analogie	36
4.6.2	La transformation structurelle	38
4.6.3	La combinaison de sources de connaissances	38
4.7	Les méthodes de la conception créatrice	38

4.8	La conception simultanée	39
4.9	Conclusion	39
5	L'Assemblage	41
5.1	Introduction	41
5.2	Définitions	42
5.3	Les processus d'assemblage	43
5.3.1	Analyse des processus d'assemblage	43
5.3.2	Typologie des tâches d'un processus d'assemblage	44
5.4	Les systèmes d'assemblage	46
5.4.1	Structure des systèmes d'assemblage	46
5.4.2	Les équipements constitutifs des systèmes d'assemblage	49
5.5	Conception des systèmes d'assemblage	51
5.6	L'analyse du problème	52
5.6.1	Les modélisations du produit	52
5.7	Caractérisation des processus d'assemblage	56
5.7.1	Détermination des processus admissibles	56
5.7.2	Sélection des processus	57
5.7.3	Représentation des processus d'assemblage	57
5.8	La caractérisation des équipements	62
5.9	La conception des commandes	64
5.10	Conclusion	64
II	Le système AMACIOA	67
6	Présentation générale	69
6.1	Introduction	69
6.2	Le processus de conception	70
6.2.1	L'espace fonctionnel	72
6.2.2	L'espace physique	73
6.2.3	L'utilisateur	73
6.3	Génération des espaces	75
6.4	Parcours des espaces	76
6.4.1	Le recuit simulé	76
6.4.2	Dynamique	78
6.5	Conclusion	79
7	Le module d'initialisation	81
7.1	L'arbre des références opératoires	81
7.1.1	Exemple	83
7.2	Le graphe de précedence	84
7.2.1	Exemple	85
7.3	Les processus d'assemblage	85
7.3.1	Détermination du graphe ET/OU	86
7.3.2	Exemple	87
7.4	Le découpage fonctionnel	87
7.4.1	Les tâches d'assemblage	89
7.4.2	Les opérations d'assemblage	89
7.4.3	Les fonctions élémentaires	90

TABLE DES MATIÈRES	11
7.5 Conclusion	90
8 Le module de sélection	93
8.1 La population d'agents	93
8.2 Dynamique du SMA de sélection	94
8.2.1 Sélection d'un processus d'assemblage	95
8.2.2 Parcours de l'espace	95
8.3 Conclusion	97
9 Le module de conception	99
9.1 La population d'agents	99
9.2 Dynamique du SMA de conception	100
9.2.1 Détermination d'un flot d'assemblage	102
9.2.2 Parcours de l'espace physique	103
9.3 Conclusion	105
10 Exemple et Évaluation	107
10.1 Introduction	107
10.2 Module d'initialisation	108
10.2.1 Analyse du problème	108
10.2.2 Caractérisation des processus d'assemblage	110
10.2.3 Exemple	110
10.2.4 Evaluation	117
10.3 Module de sélection	118
10.3.1 Exemple	119
10.3.2 Evaluation	132
10.4 Module de conception	132
10.4.1 Exemple	133
10.4.2 Evaluation	143
10.5 Conclusion	144
11 Conclusion	145
A Agents du module de sélection	155
A.1 Les agents-tâches	155
A.1.1 Objectifs	155
A.1.2 Accointances	155
A.1.3 Mémoire locale	155
A.1.4 Comportement	155
A.2 L'agent-processus	156
A.2.1 Objectifs	156
A.2.2 Accointances	157
A.2.3 Mémoire locale	157
A.2.4 Comportement	157
B Agents du module de conception	161
B.1 Les agents-tâches	161
B.1.1 Objectifs	161
B.1.2 Accointances	161
B.1.3 Mémoire locale	161
B.1.4 Comportement	162

B.2	Les agents-postes	164
B.2.1	Objectifs	164
B.2.2	Accointances	165
B.2.3	Mémoire locale	165
B.2.4	Comportement	166
B.3	Les agents-cellules	171
B.3.1	Objectifs	171
B.3.2	Accointances	172
B.3.3	Mémoire locale	172
B.3.4	Comportement	172
B.4	L'agent-flot	176
B.4.1	Objectifs	176
B.4.2	Accointances	176
B.4.3	Mémoire locale	177
B.4.4	Comportement	177
C	Difficulté d'une tâche	181
D	Détermination des équipements	183
D.1	Création des ressources génériques	183
D.2	Création d'un poste	184
D.3	Création de l'flot	185

Chapitre 1

Introduction

Depuis quelques années, les industriels ont dû élargir leur offre de produits pour répondre aux exigences du marché: ils cherchent à offrir des produits personnalisés dont la durée de vie devient de plus en plus courte, une génération de produits étant rapidement remplacée par une autre, plus innovatrice.

Ceci a amené à réviser le processus de conception de ces produits et de leurs moyens de production, avec pour seul mot d'ordre la flexibilité: flexibilité face aux demandes du marché, flexibilité des moyens de production pour traiter les variantes de produits, flexibilité du produit au cours du temps, etc. Cette flexibilité, bien que permettant de garder des parts de marché, a un coût qui se doit de rester minimal si l'entreprise veut rester compétitive.

Les systèmes d'assemblage n'échappent pas à cette règle et leur conception s'en est trouvée compliquée: l'assemblage recouvrant l'ensemble des problèmes liés à la dernière étape de la production industrielle, le système d'assemblage est la matérialisation de toutes les décisions prises par les concepteurs. Les performances du système d'assemblage (et les coûts de production qui en découlent) sont les conséquences directes de ces décisions.

La combinatoire liée à la conception des systèmes d'assemblage n'a donc cessé d'augmenter, et il devient de plus en plus difficile pour les concepteurs d'installations de trouver un compromis satisfaisant entre technologie et coût garantissant un résultat rentable.

Le système AMACIOA (Approche Multi-Agents de la COncption d'Instal-lations d'Assemblage), résultant des travaux de recherche effectués, et présenté dans ce mémoire, est un outil d'aide à la conception des systèmes d'assemblage.

La contribution de ce système réside dans l'utilisation de systèmes multi-agents réactifs, basés sur le modèle de l'éco-résolution, pour concevoir les systèmes d'assemblage. Ils sont utilisés pour parcourir l'espace de recherche que constitue la combinatoire du problème. Cette approche offre les avantages de pouvoir gérer plus facilement les interdépendances entre les paramètres du problème, et de permettre une certaine flexibilité face aux perturbations.

De plus, il s'appuie sur une nouvelle modélisation du produit, permettant de réduire l'espace de recherche, tout en garantissant de rester dans l'espace des solutions.

Le présent mémoire s'articule en deux parties:

- la partie I regroupe l'état de l'art et la présentation des notions relatives

aux divers aspects du problème, tel qu'il a été abordé. Elle se subdivise en quatre chapitres:

- le chapitre 2 présente l'approche multi-agents et plus particulièrement le modèle de l'éco-résolution.
 - le chapitre 3 présente les notions et les méthodes relatives aux problèmes combinatoires.
 - le chapitre 4 présente la formalisation et les méthodes de résolution de la conception en général.
 - le chapitre 5 présente le domaine de l'assemblage du point de vue de la conception des systèmes d'assemblage.
- la partie II est une présentation du système AMACCOIA pour la conception des systèmes d'assemblage. Elle se subdivise en cinq chapitres:
 - le chapitre 6 est une présentation générale du système. Il introduit la façon dont le problème de la conception des systèmes d'assemblage a été abordée, et les solutions qui lui ont été données.
 - le chapitre 7 présente en détails le module d'initialisation du système AMACCOIA: la modélisation du produit et la détermination des processus d'assemblage.
 - le chapitre 8 présente le module de sélection du système AMACCOIA. C'est un système multi-agents chargé de la sélection des processus d'assemblage.
 - le chapitre 9 présente le module de conception du système AMACCOIA. C'est un système multi-agents chargé de la caractérisation des équipements du système d'assemblage conçu.
 - le chapitre 10 est une présentation et une évaluation, au travers de l'exemple de la valve de vélo, des résultats obtenus.

Partie I
Etat de l'art

Chapitre 2

L'approche multi-agents

2.1 Introduction

La résolution de beaucoup de problèmes réels n'est possible que par les efforts conjugués d'un ensemble de personnes: la construction d'un immeuble nécessite un architecte, des maçons, des charpentiers, des électriciens; la conception d'un produit n'est possible que par l'interaction des différents services de l'entreprise; un match de football ne peut être gagné que par la coordination et l'organisation de l'équipe de joueurs.

Chaque personne est chargée d'une tâche spécifique mais ne possède pas les ressources suffisantes (connaissances, compétences) pour résoudre, individuellement, le problème. Une solution est obtenue par la coopération, la coordination et parfois les conflits, découlant des interactions entre les personnes du groupe, avec pour effet d'améliorer la contribution de chacun à la résolution du problème global.

Deux principes ressortent de ces faits:

- il est plus facile d'aborder un problème complexe en le *divisant* en sous-problèmes simples
- les résultats de l'*interaction* d'un ensemble d'entités constituent plus que la somme des entités prises séparément.

L'approche multi-agents cherche à tirer parti de ces principes en partageant et en distribuant entre plusieurs entités l'ensemble des connaissances, le raisonnement et la décision, liés à la résolution d'un problème. Chacune des entités est spécialisée dans un sous-domaine du domaine initial, et est dotée de moyens d'interaction avec les autres entités, qui lui permettent de gérer les interdépendances entre son sous-domaine et les autres.

Ces entités peuvent être intelligentes comme dans les systèmes multi-experts, une entité étant un expert. Elles peuvent également être très simples, l'intelligence du système étant liée au comportement global de l'ensemble des entités.

De telles entités sont appelées *agents* et les systèmes correspondants, *systèmes multi-agents*.

Le but de ce chapitre est de présenter les différentes notions liées aux agents et aux systèmes multi-agents [8, 2, 79, 61, 74, 20, 19]: les définitions, les types

d'agents, les modes d'interaction, et plus particulièrement, le modèle de l'éco-résolution.

2.2 Définitions

La notion d'agent n'est pas clairement définie et la littérature propose plusieurs définitions dont la terminologie s'apparente aux réseaux de neurones, aux systèmes répartis, aux automates cellulaires ou encore aux systèmes biologiques ou sociaux. Parmi celles-ci, J. Ferber donne la définition suivante d'un agent [25]:

“On appelle *agent* une entité physique ou abstraite qui est capable d'agir sur elle-même et sur son environnement, qui ne dispose que d'une représentation partielle de cet environnement, qui peut communiquer avec d'autres agents, qui poursuit un objectif individuel, et dont le comportement est la conséquence de ses observations, de ses connaissances, de ses compétences et des interactions qu'il peut avoir avec d'autres agents et l'environnement.”

Un *système multi-agents* est, par extension, un ensemble d'agents en interaction au sein d'un environnement.

Plusieurs notions ressortent de ces définitions: l'environnement, l'interaction, les objectifs, les connaissances et le comportement

L'*environnement* d'un agent est l'ensemble des entités qui lui sont extérieures, et au milieu duquel il agit. Ces entités sont partagées en deux catégories:

- les autres *agents*: les autres agents avec lesquels un agent *sait* être en relation sont appelés ses *accointances*. Les agents qui ne font pas partie des *accointances* sont alors traités comme des objets.
- les *objets*: les objets peuvent être, comme les agents, physiques ou abstraits ¹. Leur distinction par rapport aux agents est soit qu'ils sont inanimés, soit qu'ils n'ont pas d'objectifs si ils sont actifs (un objet ne sait pas pourquoi il agit).

L'*interaction* d'un agent avec son environnement est basée sur ses comportements, actions par lesquelles il peut modifier son environnement, et sur ses capacités de communication.

Les *objectifs*, propres à chaque agent, sont les raisons qui le poussent à agir, et les *moyens* lui permettant de mesurer le bénéfice d'une action.

Les *connaissances* d'un agent regroupent sa représentation de l'environnement, des informations sur lui-même, et des informations sur le problème qu'il a à résoudre.

Les trois notions d'objectif, de connaissances et de comportement sont liées: les objectifs permettent la sélection d'un comportement, alors que les connaissances permettent sa bonne exécution. Toutes trois sont propres à chaque agent. Ces notions permettent de classer les agents en deux catégories présentées dans les sections suivantes: les *agents cognitifs* et les *agents réactifs*.

¹Remarque: la littérature considère souvent les objets comme étant les seules entités constitutives de l'environnement. Nous lui préférons la définition donnée ci-dessus qui semble plus générique.

2.3 Les agents cognitifs

Un agent cognitif (ou *gros agent*, ou *expert*) est un agent "intelligent" dans le sens qu'il est capable de résoudre par lui-même des problèmes complexes.

Le raisonnement d'un agent cognitif s'appuie sur une base de connaissances assez importante qui regroupe non seulement sa représentation de l'environnement, les informations sur lui-même, et sur le problème qu'il a à résoudre, mais aussi une représentation des compétences de ses accointances. Cette base de connaissances forme les *croyances* de l'agent cognitif. Une croyance est une connaissance qui dépend du temps (elle peut être éphémère ou devenir erronée) et de son contexte. La mise à jour et l'utilisation des croyances par l'agent au cours du raisonnement, se basent essentiellement sur des logiques non monotones [29].

L'*intentionnalité* est la base de l'autonomie d'un agent cognitif: elle est définie comme étant le but persistant d'accomplir une action dans le présent ou dans le futur. L'agent doit relier ses objectifs, ses compétences, ses connaissances, ses engagements afin d'aboutir dans ce qu'il entreprend [12]. Ainsi, les comportements individuels d'un agent cognitif sont complexes. Ils sont liés à l'état de ses croyances, à ses objectifs et à des plans explicites, s'appuyant sur l'historique de l'agent et résultant de ses capacités de raisonnement.

Le comportement social d'un agent cognitif au sein d'un système multi-agents s'appuie sur ses capacités de communication et de coopération.

La communication est le moyen par lequel l'agent peut transmettre et recevoir de l'information qui permettra de résoudre les conflits, de participer à la résolution d'un problème, de synchroniser des actions, etc. Les principaux modèles de la communication sont issus de la théorie des actes de langage [60] et permettent un traitement logique de la communication en tant qu'acte.

Les agents cognitifs coopèrent par échange d'information ou par sous-traitance des buts. Le principal moyen de coopération est la négociation: c'est l'activité qui consiste à échanger des informations afin d'aboutir à un compromis mutuellement acceptable par les parties concernées. La négociation peut porter sur un conflit, sur la répartition des tâches entre les agents, etc.

Les systèmes multi-agents cognitifs cherchent ainsi à résoudre un problème par la coopération et la coordination d'un ensemble d'agents aux capacités de raisonnement élevées, et dotés de moyens de négociation. Comme image d'un système multi-agents cognitif, nous pouvons prendre, par exemple, un groupe d'experts travaillant en commun à la conception d'un produit.

2.4 Les agents réactifs

Un agent réactif (ou *petit agent*) est un agent peu "intelligent" possédant des comportements de type réflexe.

A l'inverse des agents cognitifs, les agents réactifs n'ont qu'une base de connaissance et une capacité de raisonnement limitées. Leurs comportements sont basés sur la réaction à des stimulations: selon les signaux qu'émet l'environnement ou selon ce qu'il perçoit de son environnement, l'agent déclenche un comportement adapté, dépendant de ses capacités de raisonnement et de ses objectifs. Ainsi le comportement d'un agent réactif ne s'appuie ni sur une explication des intentions, ni sur l'élaboration de plans.

Un agent réactif n'a pas, à proprement parler, de comportement social: sa réaction à une stimulation de l'environnement a pour conséquence le déclenchement de réactions par les autres agents, qui vont elles-mêmes déclencher d'autres réactions, etc.

Les systèmes multi-agents réactifs partent de l'hypothèse que de l'interaction d'un grand nombre d'agents simples peuvent *émerger* des organisations complexes [28]. Une illustration de cette hypothèse est la fourmilière: chaque fourmi est relativement simple et ne dispose que d'un ensemble réduit de comportements, pourtant la fourmilière fonctionne comme si elle disposait d'un organe capable de prendre des décisions et de donner des ordres aux fourmis [37].

Les systèmes multi-agents réactifs cherchent ainsi à résoudre un problème par l'identification d'une solution possible à un état de la population d'agents, état obtenu par l'auto-organisation des agents résultant de la dynamique des réactions en chaîne des agents.

2.5 Les modes de communication

Il existe trois modes de communication utilisés dans les systèmes multi-agents:

la communication par l'environnement: les agents marquent l'environnement par des traces (des signaux) de leur présence et de leurs comportements. Ces traces peuvent alors être perçues par les autres agents. Ce mode de communication, qualifié de communication non intentionnelle, est surtout utilisé dans les systèmes multi-agents réactifs.

la communication par partage d'information: tous les agents communiquent au travers d'un espace commun qui contient à tout moment les informations nécessaires à la résolution du problème. Cet espace commun, appelé *tableau noir* (blackboard) est généralement divisé en plusieurs zones, de telle sorte que chaque agent n'a en fait accès qu'à certaines d'entre elles en lecture ou en écriture. Un dispositif de contrôle, intégré au tableau noir, gère les conflits d'accès entre les agents. Ce mode de communication est surtout utilisé dans les systèmes multi-agents cognitifs.

la communication par envoi de messages: chaque agent communique directement avec ses accointances, sans passer par un tableau noir. La communication peut se faire point à point entre l'émetteur du message et un destinataire connu, ou par diffusion du message à un ensemble de destinataires (broadcast). Ce mode de communication est utilisé aussi bien dans les systèmes cognitifs, que réactifs.

2.6 L'éco-résolution

L'éco-résolution est une approche multi-agents de résolution de problème qui a été proposée par J. Ferber [25]. Elle consiste à définir une population d'agents réactifs, appelés *éco-agents* ayant des comportements simples (agression, fuite, etc.) et dont le seul objectif est la satisfaction. Le comportement global résultant de l'interaction des éco-agents permet d'atteindre un état du système qui peut être identifié comme une solution du problème.

Il s'agit en d'autres termes d'obtenir, à partir d'un comportement individuel basé sur des connaissances individuelles et la recherche de satisfaction, un comportement collectif allant dans le sens d'une satisfaction générale maximale.

Un problème multi-agents, en termes d'éco-résolution, est défini par quatre éléments:

une population d'agents: chaque agent est caractérisé par ses accointances, une mémoire locale (comprenant l'état interne de l'agent et les informations propres) et des comportements. Les comportements ne peuvent avoir que les effets suivants: mise à jour de la mémoire locale, envoi de messages, création éventuelle de nouveaux agents.

une structure de communication: le mode de communication se faisant par envoi de messages, il peut être organisé par un tour de table, par la communication à la demande, etc. La structure de communication définit également les types de messages échangés (point à point, diffusion)

un critère de terminaison: le critère de terminaison est nécessaire pour savoir si le système a atteint l'état désiré. Cela est dû au fait que l'on ne dispose pas de point de vue global.

des lois de fonctionnement: ces lois sont dictées par les principes suivants:

- le principe d'autonomie: un agent ne manipule que sa mémoire locale et ne communique que par envoi de messages avec ses accointances.
- le principe de localité: le comportement d'un agent ne dépend que de ses croyances, c'est-à-dire que des informations contenues dans sa mémoire locale ou transmises par envoi de messages.
- le principe d'économie: si un agent a plusieurs actions possibles, il exécutera celle qui lui coûtera le moins. Par exemple, entre plusieurs déplacements, un agent choisira le plus court.
- le principe d'effacement d'informations: la remise en cause d'informations de même importance porte sur les plus anciennes, les plus récentes étant plus fiables et plus proches de la vérité.
- le principe de maximisation du plaisir: un agent cherche toujours à se placer dans un état pour lequel le maximum de ses contraintes sont satisfaites.

Le comportement d'un éco-agent est généralement constitué d'actes réflexes très simples (*tropisme*), et guidé par la recherche de satisfaction. Un tropisme peut s'exprimer par le quintuplet $\langle C_{sat}, C_{rjt}, R_s, f_{cE}, I \rangle$ avec:

- C_{sat} , est l'ensemble des conditions de satisfaction locales décrivant les états stables pour l'agent, en fonction de ses croyances.
- C_{rjt} , est l'ensemble des conditions de rejet locales décrivant les situations que rejette (cherche à fuir) l'agent.
- R_s , est l'ensemble des réactions de survie de l'agent. Une réaction de survie est un ensemble d'actions qui, selon les circonstances, tendent à rapprocher l'agent de ses conditions de satisfaction. Ces actions peuvent être: des

actions de fuite (exécutées lorsqu'une condition de rejet est atteinte), des actions d'agression (exécutées lorsqu'un autre agent gêne la réalisation des conditions de satisfaction), des actions de satisfaction (exécutées pour réaliser les conditions de satisfaction).

- f_{cE} , est une fonction locale du coût énergétique du tropisme qui permet de l'évaluer dans le cadre du principe d'économie.
- I , est l'ensemble des informations diffusées sous forme de messages aux accointances concernées.

L'éco-résolution a déjà montré son élégance et son efficacité sur des problèmes classiques en intelligence artificielle (le taquin [24], le monde des cubes [25]), mais aussi sur des problèmes moins "triviaux" (la résolution des problèmes de satisfaction de contraintes [31], les échecs [23]).

2.7 Conclusion

L'approche multi-agents est une méthode de résolution de problèmes basée sur l'interaction d'un ensemble d'entités.

Chaque agent est responsable de la résolution d'un sous-problème et possède une certaine autonomie, sous la forme d'un pouvoir de raisonnement et de décision, basé sur ses connaissances et guidé par un objectif qui lui est propre. Ses capacités d'interaction sont constituées de ses comportements (par lesquels il influence son environnement) et de ses capacités de communication.

Les systèmes multi-agents cognitifs, construits sur des agents "intelligents" capables de résoudre par eux-mêmes des problèmes complexes, obtiennent une solution par la coopération et la coordination, résultant des négociations entre les agents.

Les systèmes multi-agents réactifs, construits sur des agents simples ayant des comportements de type réflexe, obtiennent, quant à eux, une solution par l'identification de celle-ci à un état de la population d'agents, état obtenu par l'auto-organisation des agents résultant de leurs interactions.

Chapitre 3

Les problèmes combinatoires

3.1 Introduction

Beaucoup de problèmes, aussi bien pratiques que théoriques, trouvent une solution dans une configuration d'un ensemble d'éléments: une partie d'échecs s'achève lorsque la configuration des pièces sur l'échiquier empêche le mouvement du roi adverse; l'allocation de ressources configure un ensemble de tâches, de manière à ce qu'elles soient réalisées par un ensemble de ressources; une solution d'un système d'équations est un vecteur regroupant les valeurs des variables qui satisfont l'ensemble d'équations. La nature des solutions, en tant que combinaison d'éléments, désigne ces problèmes comme les *problèmes combinatoires*.

Toutes les combinaisons possibles des éléments d'un problème combinatoire ne constituent pas forcément une solution du problème: pour être solution, une combinaison doit répondre à un certain nombre de *contraintes*: seule une configuration de pièces qui menace le roi adverse sans possibilité de défense permet de gagner la partie; la configuration des tâches allouées ne doit pas dépasser les capacités de chacune des ressources et respecter les contraintes de précedence pour leur exécution; les valeurs des composantes du vecteur doivent satisfaire toutes les équations du système.

La résolution d'un problème combinatoire consiste donc à rechercher une configuration qui résolve toutes les contraintes du problème. C'est généralement un processus itératif, dont une étape consiste à modifier une configuration courante en une nouvelle configuration plus "proche" de la solution: jouer un coup aux échecs; permuter deux tâches sur une ressource; modifier la valeur des composantes du vecteur. Le processus de résolution constitue un *chemin* (passage d'une configuration à l'autre) au sein de l'ensemble des combinaisons possibles.

Comme, en général, il n'y a pas qu'une seule configuration qui soit solution du problème, la résolution de certains problèmes combinatoires va chercher à trouver non seulement une solution, mais la solution la "meilleure" possible au vu de l'objectif à atteindre: le joueur d'échecs cherche le mat en moins de coups possible; la configuration des tâches allouées doit minimiser les changements d'outils pour chacune des ressources; la norme du vecteur, solution du système

d'équations, doit être minimale. Ainsi la résolution d'un problème combinatoire est souvent un processus d'*optimisation*, dont les critères peuvent porter aussi bien sur les solutions que sur le parcours.

Le but de ce chapitre est de présenter succinctement les notions inhérentes aux problèmes combinatoires et à leur résolution. Ces notions sont tirées principalement de [52] et de [27].

3.2 Définitions

Un problème combinatoire est défini par le couple $\langle V, C \rangle$ avec:

- V , l'ensemble des variables caractéristiques du problème: $V = \{v^1, \dots, v^n\}$. Chaque variable $v^i \in V$ est caractérisée par l'ensemble, $D_{def}^{v^i}$, des valeurs qu'elle peut prendre.
On appelle *état*, une combinaison possible de valeurs, prises dans les ensembles de valeurs $D_{def}^{v^i}$, pour chacune des variables $\in V$. Un état est donc déterminé par les valeurs attribuées à chaque variable.
- C , l'ensemble des *contraintes* du problème: $C = \{c^1(V), \dots, c^m(V)\}$. Une contrainte est une relation sur les ensembles des valeurs des variables $\in V$ représentant une interdépendance entre les variables qui doit être respectée.

L'ensemble de tous les états possibles pour les variables $\in V$, est appelé l'*espace d'états* du problème, et il est noté E_e . L'espace d'états est soit continu soit discret, selon que les ensembles des valeurs $D_{def}^{v^i}$ des variables soient continus ou discrets: dans le cas de la résolution de système d'équations, \mathbb{R}^n est un espace d'états continu, l'ensemble des valeurs que peut prendre chaque composante du vecteur étant égal à \mathbb{R} ; l'espace d'états des échecs est discret, les positions que peut avoir chacune des pièces étant les 64 cases de l'échiquier.

Un état $e_s \in E_e$ des variables $\in V$, tel que les m contraintes $\in C$ soient satisfaites, est appelé une *solution* du problème. L'ensemble des solutions du problème, est appelé l'*espace des solutions*, et il est noté E_s .

Il existe deux classes de problèmes combinatoires, selon la nature de la solution recherchée: les *problèmes de satisfaction de contraintes* et les *problèmes d'optimisation combinatoire*.

- la résolution d'un problème de satisfaction de contraintes consiste uniquement à rechercher un état $e_s \in E_s$, qui satisfasse les contraintes.
- la résolution d'un problème d'optimisation combinatoire consiste à rechercher un état $e_{opt} \in E_s$, tel que les contraintes soient satisfaites et que $f_c(e_{opt}) \leq f_c(y)$ pour tout $y \in E_e$ (ou $f_c(e_{opt}) \geq f_c(y)$ pour tout $y \in E_e$, si on cherche à maximiser $f_c(e)$).

$f_c(e)$ est la *fonction coût* (ou encore la *fonction objectif*) que l'on cherche à optimiser. La fonction coût, caractéristique du problème d'optimisation combinatoire, fait correspondre à chaque état $e \in E_e$, une valeur:

$$f : E_e \longrightarrow \mathbb{R}^1 \quad (3.1)$$

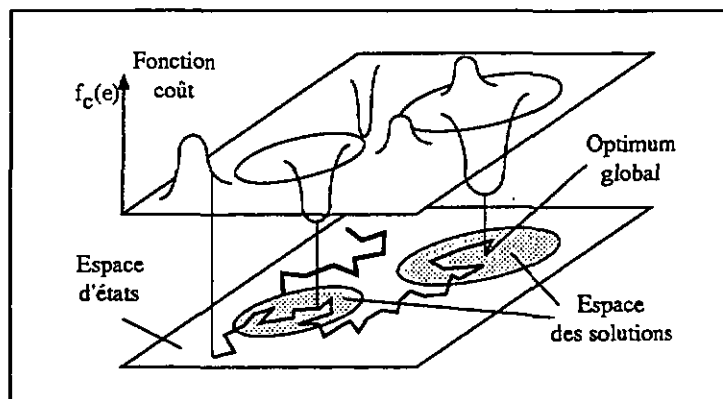


Figure 3.1: Les éléments caractéristiques des problèmes combinatoires. En gras, parcours possible d'un problème de satisfaction de contraintes s'arrêtant dès que l'espace des solutions est atteint. En fin, parcours possible d'un problème d'optimisation combinatoire, s'appuyant sur les valeurs de la fonction coût et s'arrêtant lorsque l'optimum global est atteint.

L'état e_{opt} est appelé l'*optimum global* du problème ou, quand il ne peut y avoir confusion, simplement l'*optimum*.

3.3 L'espace de recherche

L'espace d'états peut être appelé aussi l'espace de recherche et il est alors noté E_r . La notion d'espace de recherche est liée à la résolution du problème combinatoire, dans le sens que l'on va se déplacer dans l'espace d'états à la recherche de la solution du problème.

L'espace de recherche est un espace muni d'une (ou plusieurs) fonction(s) d'*adjacence*. Une fonction d'adjacence est une fonction qui associe un état $e \in E_r$ à son ensemble $N(e)$ (appelé aussi *voisinage*) des états $\in E_r$ qui lui sont "proches" dans un certain sens. La fonction d'adjacence fait correspondre:

$$N : E_r \longrightarrow \Pi(E_r), \text{ l'ensemble des parties de } E_r \quad (3.2)$$

Le choix de la fonction d'adjacence dépend de la méthode de déplacement dans l'espace et de la structure de l'espace de recherche: si $E_r = \mathbf{R}^n$, l'ensemble des points situés à une distance euclidienne donnée constitue un voisinage naturel; le fait de jouer un coup aux échecs constitue une fonction d'adjacence.

Un sous-espace particulier de l'espace de recherche est l'*espace des solutions possibles*, noté $E_{r,s}$. L'espace des solutions possibles n'est autre que l'espace des solutions. C'est ce sous-espace que la résolution du problème va chercher à atteindre, dans le cas des problèmes de satisfaction de contraintes, ou à parcourir jusqu'à l'optimum, dans le cas des problèmes d'optimisation combinatoire.

3.4 Le parcours de l'espace

Un algorithme de parcours d'un espace de recherche dépend de deux aspects:

- la garantie d'atteindre une solution (ou *complétude* de l'algorithme): en effet, selon la méthode utilisée, la trajectoire du parcours peut se retrouver dans un état de l'espace de recherche qui ne permet plus d'atteindre l'espace des solutions possibles ou l'optimum global.
- le temps nécessaire pour parvenir à la solution: ce temps, qui dépend aussi de la taille du problème, est une mesure de la complexité des algorithmes de parcours. On cherche, en général, à ce que ce temps soit une fonction polynomiale de la taille du problème: n, n^2, n^3, \dots, n étant une grandeur caractéristique de la taille du problème.

En général, que ce soit pour les problèmes de satisfaction de contraintes ou pour les problèmes d'optimisation combinatoire, le parcours de l'espace de recherche est un processus itératif, dont une étape consiste:

1. à générer (ou à sélectionner) au moyen de la fonction d'adjacence le voisinage de l'état courant e_{crt} , c'est-à-dire l'ensemble des états $N(e_{crt})$ "proches" de e_{crt} .
2. à choisir parmi les états de $N(e_{crt})$, l'état qui deviendra le nouvel état courant e_{crt} . Ce choix est fait de manière à se rapprocher de la solution recherchée. Il est généralement guidé par les contraintes à résoudre (par exemple, choisir l'état qui satisfait le plus de contraintes) ou par des heuristiques dépendantes du domaine. Dans le cas des problèmes d'optimisation combinatoire, il dépend de la fonction coût.

Il arrive aussi qu'une partie de la procédure de choix soit directement incluse dans la fonction d'adjacence, avec pour conséquence directe de ne générer que les états $\in N(e_{crt})$ les plus prometteurs.

La complétude de cet algorithme pose généralement peu de problèmes dans le cas des problèmes de satisfaction de contraintes: la garantie d'atteindre l'espace des solutions possibles peut être obtenue par la fonction d'adjacence et la procédure de choix du nouvel état courant.

Par contre, dans le cas des problèmes d'optimisation combinatoire, dont le but est de trouver l'optimum global du problème, la complétude est remise en cause par les optima locaux.

La solution simple consistant au parcours systématique de tout l'espace de recherche (qui garantirait de trouver l'optimum global) est généralement impossible: par exemple, il y a $1.308 \cdot 10^{89}$ manières de disposer les 32 pièces d'un jeu d'échecs sur l'échiquier, ce qui prendrait à peu près $4.1473 \cdot 10^{81}$ années pour les parcourir, à raison d'une configuration par seconde.

Les *algorithmes exacts*, tels que l'algorithme du Simplex, l'algorithme du Branch and Bound [52] ou ceux développés pour les problèmes de programmation linéaire, garantissent d'obtenir l'optimum global. Par contre ces algorithmes ne sont pas des algorithmes à temps polynomial, et ne peuvent s'appliquer qu'aux problèmes de taille relativement petite.

Les *algorithmes stochastiques*, tels que la méthode Tabou, les algorithmes génétiques ou le recuit simulé, ne garantissent pas l'obtention de l'optimum

global (sauf en théorie pour des temps de calcul infinis), mais l'obtention d'une solution sub-optimale "satisfaisante".

La méthode Tabou [33, 32, 18] génère à chaque pas, sur la base de l'état courant, un ensemble fini de nouveaux états de l'espace de recherche. La taille de l'ensemble généré dépend du problème et de la structure de l'espace de recherche. La fonction coût est calculée pour chaque état généré et le meilleur état de l'ensemble devient alors le nouvel état courant. Cet état est marqué (Tabou) pour qu'il ne soit pas exploré à nouveau dans un avenir trop proche.

Les algorithmes génétiques [36, 49] sont basés sur le paradigme de la génétique biologique. Un état du problème est modélisé sous la forme d'un "chromosome" (chaîne d'éléments significatifs). A chaque pas, des chromosomes parents sont croisés (en utilisant des opérateurs de mutation et de crossing over) pour obtenir une nouvelle population de chromosomes. Cette population est alors évaluée sur la base de la fonction coût, et les meilleurs chromosomes deviennent alors les parents de la génération suivante.

Le recuit simulé [45, 1] est basé sur le paradigme du recuit thermique ¹. Le nouvel état généré est comparé à l'état courant: la comparaison détermine une probabilité d'acceptation du nouvel état comme nouvel état courant. La probabilité d'acceptation dépend de la différence des valeurs de la fonction coût dans les deux états et d'un paramètre appelé la température. Si le nouvel état est accepté, il devient l'état courant. Le recuit simulé peut se prêter à une implantation distribuée [31, 39].

Parmi les algorithmes stochastiques, le recuit simulé permet dans une certaine mesure d'échapper aux optima locaux: en effet la probabilité d'acceptation d'un état dont la valeur de la fonction coût est moins bonne que celle de l'état courant, n'est pas nulle. En permettant l'acceptation de tels états, il est possible de quitter un optimum local pour passer dans une zone de l'espace de recherche qui peut contenir l'optimum global.

Le parcours de l'espace de recherche par les algorithmes stochastiques est arrêté après un certain nombre de pas (dépendant du problème), ou lorsqu'une solution "satisfaisante" (dépendante par exemple d'une valeur seuil de la fonction coût) a été atteinte. La mise en oeuvre de ces algorithmes n'est pas immédiate, et nécessite une phase de réglage des différents paramètres (fonction coût, fonction d'adjacence, critères d'arrêt) pour être efficace.

3.5 Problèmes combinatoires et approche multi-agents

L'approche multi-agents constitue une méthode intéressante et naturelle pour la résolution de certains problèmes combinatoires. En effet, un problème combinatoire trouve une solution dans une combinaison de valeurs qui satisfait un ensemble de contraintes, et l'approche multi-agents apporte une solution à un problème en gérant les interdépendances entre les sous-problèmes du problème global par l'interaction d'un ensemble d'agents.

En identifiant, par exemple, les valeurs du problème combinatoire aux agents, et les contraintes aux interdépendances, il est alors possible de générer les états et le parcours de l'espace par la dynamique des agents.

¹Le recuit simulé est décrit plus en détails au point 6.4.1.

Comme exemples d'application de l'approche multi-agents à des problèmes combinatoires, nous pouvons citer:

- le système MASC [31] appliqué au problème de l'allocation de ressources, et basé sur le modèle de l'éco-résolution. Les agents sont constitués des tâches et des ressources. Une tâche cherche à se satisfaire en se plaçant sur une ressource, et une ressource en maximisant le nombre de tâches qu'elle peut exécuter. Un instantané de la situation des tâches sur les ressources constitue un état de l'espace du problème, alors que la dynamique des requêtes des tâches, et des acceptations et rejets par les ressources, permet de se déplacer dans l'espace, en passant d'un état à un autre. Le parcours s'arrête lorsque tous les agents sont satisfaits. La procédure d'acceptation ou de rejet des tâches par les ressources est contrôlée par un recuit simulé distribué, chaque ressource ayant son propre recuit.
- le système présenté dans [2] traite du problème du voyageur de commerce, qui consiste à trouver le chemin le plus court reliant un ensemble de villes. Au départ, les agents sont répartis dans chacune des villes. Chaque agent va alors effectuer un cycle, en passant par toutes les villes une et une seule fois. A la fin de son cycle, il marque chaque portion de son trajet par une certaine quantité (dépendante de la longueur de son cycle) de substance évanescence, puis il repart pour un nouveau cycle. A chaque étape de son parcours, l'agent choisit sa prochaine ville selon une certaine probabilité dépendante de la distance et de la quantité de substance se trouvant déjà sur le trajet jusqu'à cette ville. Ainsi, au fur et à mesure des cycles des agents, le cycle le plus court émerge: c'est le cycle dont les portions sont le plus marquées par la substance.

Un cycle d'un agent constitue un état de l'espace du problème, alors que la dynamique des choix et du marquage, permet le parcours et la convergence vers la solution.

3.6 Conclusion

Un problème combinatoire est défini par un ensemble de variables caractéristiques et un ensemble de contraintes sur ces variables. L'espace d'états du problème est constitué de l'ensemble des états (combinaisons de valeurs) possibles pour les variables, et l'espace des solutions est constitué des états pour lesquels les contraintes sont satisfaites.

La résolution d'un problème de satisfaction de contraintes consiste à trouver un état de l'espace des solutions. La résolution d'un problème d'optimisation combinatoire consiste à trouver l'optimum global de l'espace des solutions, état pour lequel la valeur de la fonction coût est optimale.

La résolution d'un problème combinatoire est un chemin au travers d'un espace de recherche que constitue l'espace d'états. Ce parcours s'appuie sur des fonctions d'adjacence, afin de générer des nouveaux états de l'espace de recherche, et les contraintes (ou la fonction coût) pour sélectionner le nouvel état courant. Le but est, soit un état satisfaisant les contraintes (satisfaction de contraintes), soit l'optimum (optimisation combinatoire) de l'espace des solutions possibles.

Chapitre 4

La conception

4.1 Introduction

La conception existe depuis que les conditions d'existence qu'offre le monde ne satisfont plus l'être humain. Dès lors, il a toujours cherché à améliorer ces conditions par la création d'artefacts. En concevant ces objets, il réussit à changer le monde afin de l'adapter à ses besoins.

La science traditionnelle cherche à comprendre le monde par une démarche analytique, selon laquelle elle tente de dégager les fonctions qui le gouvernent à partir d'une description du monde et des phénomènes qui l'animent. La conception, quant à elle, commence par fixer un ensemble de fonctions à réaliser, avant de produire la description d'un artefact capable de générer ces fonctions. Sa démarche est synthétique et cherche à faire le monde. En ce sens, la démarche de la conception est l'opposé de la démarche scientifique traditionnelle.

La conception apparaît partout et traite une grande variété de problèmes. Certains de ces problèmes sont issus de situations de la vie courante: concevoir un repas en utilisant les restes de la veille; concevoir un arrangement floral plaisant à l'oeil; concevoir une sorte de crochet pour récupérer un objet tombé derrière un radiateur; concevoir une barrière pour empêcher un enfant de tomber dans l'escalier. D'autres problèmes sont simples, mais plus commerciaux: concevoir un portemanteau; concevoir une lampe de bureau articulée; concevoir un casse-noisettes. D'autres problèmes encore sont beaucoup plus complexes et nécessitent le travail et la coordination de plusieurs concepteurs: concevoir un avion; concevoir un satellite de télécommunication; concevoir un système d'assemblage.

Ces dernières décennies, la recherche a permis de dégager une formalisation de la conception, ainsi qu'un certain nombre de méthodes qui sont utilisées principalement pour offrir une aide active aux concepteurs. Le but de ce chapitre est de présenter cette formalisation et les principales méthodes utilisées: la conception est présentée comme un problème combinatoire travaillant sur deux espaces de recherche liés, *l'espace fonctionnel* et *l'espace physique*.

4.2 Typologie des problèmes de conception

La typologie des problèmes de conception présentée ci-après a été introduite par D. C. Brown et B. Chandrasekaran [7]. Bien que les limites entre classes soient assez floues pour catégoriser parfois certains problèmes, elle est reconnue par la plupart des auteurs. Cette typologie est basée sur les connaissances et les méthodes à disposition pour générer des solutions. Elle distingue trois classes de problème de conception :

- la *conception courante* (routine design) est caractérisée par le fait que les connaissances et les méthodes à disposition sont toujours suffisantes pour générer (plus ou moins) directement une solution. La conception courante peut être vue comme une succession de sélections au sein d'ensembles connus d'alternatives bien définies. Les choix en chaque point peuvent être simples, mais cela n'implique pas que le processus de conception lui-même soit simple ou que les structures physiques ainsi conçues doivent être simples.
- la *conception innovatrice* (innovative design) est caractérisée par le fait que les connaissances nécessaires sont a priori identifiées, alors que les méthodes à disposition sont lacunaires et nécessitent au préalable de combler ces lacunes pour pouvoir générer une solution. La conception innovatrice implique donc l'utilisation de nouvelles méthodes et une part de planification sur la stratégie de résolution.
- la *conception créatrice* (creative design) est caractérisée par le fait que ni les connaissances, ni les méthodes à disposition ne sont connues au préalable. Il est donc nécessaire de les établir avant de pouvoir générer une solution.

Des méthodes spécifiques à chaque classe de problème ont été développées. La majorité des systèmes existants portent sur des problèmes de conception courante et de conception innovatrice, les systèmes pour la conception créatrice n'étant encore qu'à leurs balbutiements.

4.3 Définitions

La conception est la transformation d'un ensemble donné de *fonctions* en une *structure physique* réalisable telle qu'elle produise ces fonctions. Le résultat de la conception, plutôt que la structure physique elle-même, est une *description* de celle-ci qui contienne suffisamment d'information pour sa fabrication [30, 65].

En plus de la fonctionnalité souhaitée, la structure physique conçue doit répondre à un certain nombre de *contraintes*. Les contraintes de conception peuvent porter sur [9] : les caractéristiques de la structure physique (par exemples : son poids, son coût, sa couleur, etc.), le procédé de fabrication de la structure physique (par exemple : l'imposition d'une technologie particulière), le processus de conception lui-même (par exemple : un délai d'obtention d'un premier prototype), etc.

Le terme "fonction" est pris ici dans son sens de "rôle caractéristique que joue une chose dans l'ensemble dont elle fait partie" [57], c'est-à-dire ce pourquoi

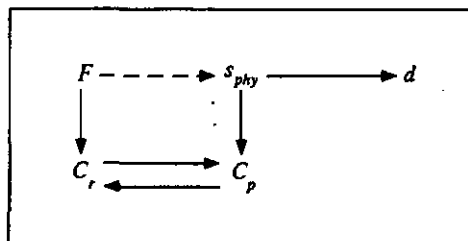


Figure 4.1: Les éléments de la conception et les relations existant entre ceux-ci. F , l'ensemble des fonctions; C_r , l'ensemble des comportements recherchés; C_p , l'ensemble des comportements productibles; s_{phy} , la structure physique; d , la description de la structure physique. Une flèche indique une transformation et une flèche traitillée, une transformation occasionnelle.

un objet est fait. Certains auteurs l'utilisent parfois dans son sens "d'action caractéristique d'un élément dans un ensemble", c'est-à-dire ce que fait un objet. Pour lever l'ambiguïté entre ces deux sens, nous appellerons *fonction*, le rôle d'un objet et *comportement*, l'action d'un objet¹.

Une fonction est un ensemble de *comportements recherchés*. Un comportement recherché est la caractérisation d'un ensemble d'objets fonctionnels (information, énergie, type d'objets physiques) et des relations qui existent entre ceux-ci. Les comportements recherchés fournissent la syntaxe selon laquelle sont exprimées les manières de réaliser les fonctions, donc ce que peuvent faire les fonctions. Par exemple, la fonction "donner l'heure" peut être composée des comportements recherchés "fournir une fonction du temps", "fixer une référence", "mesurer l'intervalle entre l'instant présent et la référence" et "visualiser la mesure" (ce qui pourra correspondre à une montre, un cadran solaire ou une clepsydre); un autre ensemble possible de comportements recherchés serait "recevoir une requête", "connaître l'heure" et "transmettre l'information" (ce qui pourra correspondre à l'horloge parlante ou à une personne à qui l'on demande l'heure).

La structure physique est un ensemble de *comportements productibles*. Un comportement productible est la caractérisation d'un ensemble d'objets physiques (les composants de la structure physique) et des relations entre ceux-ci. Les comportements productibles permettent d'exprimer à la fois l'organisation des composants au sein de la structure physique et ce que fait la structure physique. Par exemple, l'ensemble constitué d'un moteur, de son axe et d'une roue d'entraînement est caractérisé par le comportement productible de "fournir un couple".

Le *processus de conception* est la succession des transformations qui vont amener des fonctions à la description de la structure physique. Le processus de conception cherche à préciser la description des fonctions en les exprimant par les comportements recherchés, et la description des structures physiques en les exprimant par les comportements productibles, afin de rendre alors possible l'identification entre comportements recherchés et comportements productibles.

¹ Les différents points de vue sur la fonctionnalité en conception sont présentés de manière plus détaillée dans [78] par J. Winsor et K. MacCallum

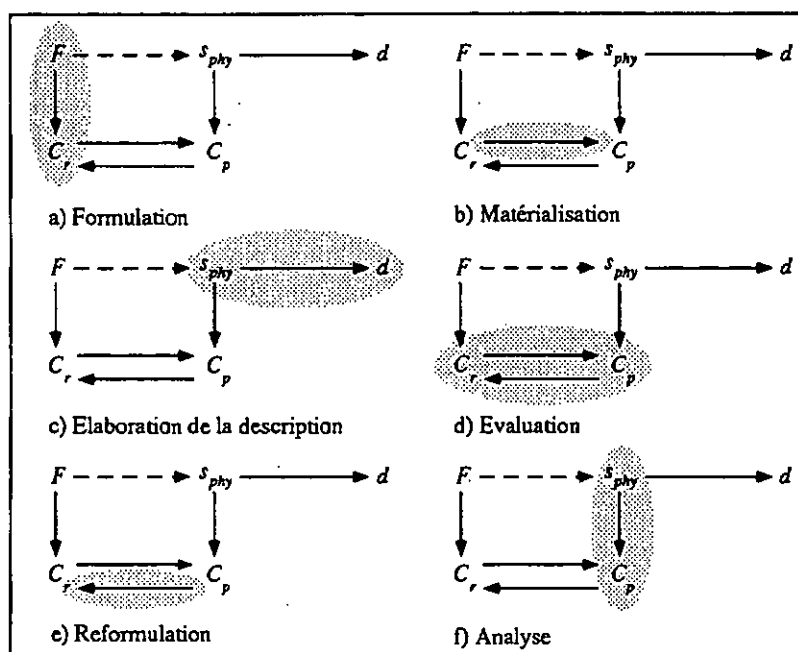


Figure 4.2: Les opérations de transformation sur les éléments de la conception

4.4 Le processus de conception

Les opérations qui constituent un processus de conception sont les différentes transformations que subissent les éléments de la conception. Ces opérations sont les suivantes [30]:

- *la formulation du problème*: c'est la transformation des fonctions en comportements recherchés, afin d'exprimer la manière dont vont être réalisées les fonctions.
- *la matérialisation*: c'est la transformation des comportements recherchés en comportements productibles. Les objets fonctionnels et leurs interactions (qui forment un comportement recherché) sont identifiés à des composants et leurs interactions (qui forment un comportement productible). La matérialisation permet alors la synthèse de la structure physique par regroupement des comportements productibles matérialisant les comportements recherchés.
- *l'élaboration de la description*: la structure physique conçue est décrite sous une forme (plans, texte) qui permette le transfert de l'information nécessaire à la production.
- *l'évaluation*: c'est la comparaison entre les comportements recherchés et les comportements productibles. Cette évaluation permet de vérifier la conformité entre la structure physique conçue et la fonctionnalité attendue.

- *la reformulation du problème*: un comportement productible peut matérialiser d'autres comportements recherchés que ceux pour lesquels il a été initialement retenu. Ceci peut amener à reformuler le problème, en termes de comportements recherchés, afin de tenir compte de ces nouvelles fonctions.
- *l'analyse de la structure*: c'est la transformation de la structure physique en ses comportements productibles. Cette opération permet d'extraire les connaissances nécessaires aux opérations de matérialisation, d'évaluation et de reformulation.

La nature de ces opérations, la manière de les réaliser au sein d'un processus de conception et les éléments sur lesquels elles portent, déterminent deux espaces de recherche distincts, mais liés: *l'espace fonctionnel* et *l'espace physique*. Un processus de conception est alors un parcours au travers de ces deux espaces.

4.4.1 L'espace fonctionnel

L'espace fonctionnel est déterminé par le fait qu'il n'y a pas qu'une seule manière de réaliser une fonction donnée: la formulation du problème en termes de comportements recherchés résulte dans l'expression abstraite de ces différentes manières.

Ainsi, un point de l'espace fonctionnel est une combinaison de comportements recherchés, décrivant de manière abstraite une façon de réaliser l'ensemble des fonctions: une telle combinaison est appelée une *spécification*. La nature de la combinaison et les comportements recherchés qui la constituent dépendent, en général, des connaissances sur le domaine d'application du problème de conception. L'ensemble des spécifications constitue l'espace fonctionnel.

L'espace fonctionnel est souvent exprimé à différents niveaux d'abstraction. Chaque niveau est obtenu par *décomposition fonctionnelle* du niveau supérieur, ce qui permet de préciser la description des spécifications. La décomposition fonctionnelle introduit une syntaxe propre à chaque niveau, constituée des éléments et de leurs relations au sein de la spécification.

4.4.2 L'espace physique

L'espace physique est déterminé par le fait qu'il n'y a pas qu'une seule manière de synthétiser une structure physique: la matérialisation des comportements recherchés en comportements productibles peut se faire de différentes façons, de même que le regroupement des comportements productibles au sein de la structure physique.

Un point de l'espace physique est une combinaison de comportements productibles formant une structure physique.

L'espace physique s'exprime aussi à différents niveaux de description. Les décompositions fixant un niveau donné peuvent être de deux types:

- *la décomposition structurelle*: cette décomposition permet de mettre en évidence l'organisation des composants au sein de la structure physique. Le niveau le plus élevé consiste dans la structure physique complète, et le niveau le plus bas dans les composants élémentaires.

- *la décomposition comportementale*: cette décomposition permet de préciser la description des comportements productibles de la structure physique.

Un niveau de description peut être aussi issu d'une combinaison de ces deux types de décomposition.

4.4.3 Les liens entre espaces physique et fonctionnel

L'espace fonctionnel et l'espace physique sont les représentations des deux aspects de la conception: ce qui doit être fait (espace fonctionnel) et comment le faire (espace physique).

Les deux espaces sont liés par les trois opérations de matérialisation, de reformulation et d'évaluation. La matérialisation permet le passage de l'espace fonctionnel à l'espace physique en déterminant l'implantation physique qui réalisera les spécifications. La reformulation permet le passage inverse, de l'espace physique à l'espace fonctionnel, en déterminant ce qu'il peut être fait, en plus des fonctions initiales, par la structure physique. L'évaluation permet de vérifier la validité d'un passage d'un espace à l'autre.

Ces opérations peuvent intervenir entre n'importe quels niveaux des deux espaces. Ainsi le parcours des espaces de recherche de la conception alterne des trajectoires dans chacun des espaces avec des passages de l'un à l'autre, pour se terminer dans l'espace physique.

4.5 Les méthodes de la conception courante

En conception courante, la plupart des méthodes utilisées peuvent être vues comme des méthodes de *décomposition - recombinaison* [47]. Ces méthodes procèdent de l'idée largement répandue qu'un problème complexe est plus facile à appréhender en le divisant en sous-problèmes simples. Ainsi dans une première étape le problème est décomposé en sous-problèmes plus simples, auxquels on recherche une solution, avant de recomposer ces solutions dans la structure physique finale.

La décomposition d'un problème devrait idéalement amener à des sous-problèmes indépendants, afin que la recombinaison des solutions partielles soit la plus implicite possible. Ce fait est rarement vérifié et peut causer des complications au moment de la recombinaison des solutions partielles dans la solution finale. De plus, il existe généralement plus d'une manière de décomposer un problème, ce qui signifie qu'une représentation de la décomposition qui permette d'exprimer ces variantes devrait exister.

La décomposition peut porter sur:

- les fonctions à réaliser. C'est une décomposition fonctionnelle jusqu'à un niveau de détail suffisant pour permettre l'identification directe entre comportements recherchés et comportements productibles.
- les connaissances du domaine. C'est une décomposition structurelle des connaissances dans les différents composants physiques sur lesquels est construite une solution au problème posé.

La décomposition du problème peut se faire a priori et le processus de conception s'appuie alors sur cette structure. C'est souvent le cas dans les domaines bien connus et n'évoluant que très peu: les systèmes construits sur ce principe sont peu flexibles et nécessitent de grosses modifications en cas de changement des connaissances. C'est pourquoi, on préfère souvent effectuer une décomposition dynamique.

4.5.1 Les méthodes non basées connaissance

Pour certains problèmes, une méthode basée connaissance est inutile et les méthodes conventionnelles peuvent être suffisantes pour les résoudre. C'est notamment le cas, lorsque le problème de conception peut être exprimé sous la forme d'un problème combinatoire (avec ou sans optimisation) sur des variables réelles, le but étant de déterminer les valeurs des paramètres caractéristiques de la structure physique: les méthodes de *programmation linéaire*, ou les méthodes présentées au chapitre 3, sont applicables. De la même manière, des problèmes de conception se prêtent très bien à d'autres méthodes standards: programmation entière, techniques d'optimisation multi-critères (comme la conception de réseaux de communication informatiques [54]), parcours de graphes ET/OU, techniques d'analyse numérique, etc.

En règle générale, la plupart de ces méthodes ne sont pas appliquées à la résolution du problème entier (sauf dans les cas simples), mais elles sont utilisées pour la résolution d'un sous-problème spécifique.

4.5.2 Les méthodes basées connaissances

La conception courante, vue comme un parcours basé connaissance de l'espace de recherche, s'appuie sur plusieurs types d'opérations de base: les opérations de *décomposition* et de *matérialisation* génèrent de nouveaux points moins abstraits de l'espace de recherche; les opérations de *résolution de contraintes* permettent soit l'élimination d'alternatives de recherche inconsistantes, soit la conversion d'une solution incorrecte en une solution acceptable; les opérations d'*optimisation* permettent la conversion d'une solution sous-optimale, selon certains critères d'optimisation, en une solution plus proche de l'optimum. Ces opérations sont souvent implantées soit sous forme de règles dont l'application requiert la conformité à une forme de données, soit sous la forme de plans (ou procédures) indexés par le type de données.

Les processus de conception basés sur ces opérations peuvent être itératifs ou non. Dans les cas non-itératifs, les connaissances ou les méthodes sont suffisantes pour permettre en plus ou moins une seule passe descendante de converger vers une solution acceptable - avec l'aide d'opérations de résolution de contraintes et l'application de connaissance de contrôle. Comme exemples de tels systèmes, on peut citer AIR-CYL [7], un système de conception pour vérins pneumatiques, VEXED [66], un système de conception pour circuits électroniques VLSI, ADES [13], un système de conception pour les systèmes de contrôle des gares ou un système de conception pour liens de communication en fibres optiques [15].

Lorsque plusieurs contraintes ou objectifs doivent être satisfaits, le processus de conception devient généralement itératif. Plusieurs formes d'itérations sont possibles:

- le backtracking: soit chronologique tel que dans le système TRANSFORM [66], un système de conception pour boîtes à vitesse automatiques, soit guidé connaissance (utilisation d'heuristiques) tel que dans les systèmes PRIDE [50], un système de conception pour le parcours du papier dans les copieurs, VT [66], un système de conception pour ascenseurs et COMIX [6], un système de conception pour agitateurs industriels.
- le hillclimbing: optimisation ou résolution de contraintes itérative jusqu'à ce que toutes les contraintes soient satisfaites ou qu'un compromis satisfaisant soit obtenu pour les critères globaux d'optimisation. Les connaissances utilisées peuvent être une fonction d'évaluation ou un ensemble d'heuristiques dépendants du domaine (comme dans CHIPPE [5], un système de conception pour circuits VLSI), fournies par l'utilisateur (DESIGNER [66], un système de conception pour interfaces graphiques).
- la restructuration du problème: à chaque itération, il est possible de modifier le problème en soi, en ajoutant de nouvelles contraintes ou spécifications et en relâchant ou durcissant d'anciennes. Ces modifications sont généralement laissées à la responsabilité de l'utilisateur tel que dans les systèmes BIOSEP [66], un système de conception pour les processus de purification de protéines, et WRIGHT [66], un système de conception pour layouts 2D.

4.6 Les méthodes en conception innovatrice

Dans le cas de la conception innovatrice, les méthodes décrites ci-dessus ne peuvent être appliquées; ceci est dû à un manque de connaissances pour générer un nouveau point de l'espace de recherche. La première cause de ce manque est souvent que les connaissances nécessaires ne sont pas forcément sous une forme directement applicable. Une seconde raison du manque de connaissances vient de l'impossibilité de stocker la grande quantité de connaissances spécifiques nécessaires pour traiter de manière adéquate tous les aspects variables de la conception (variation de la spécification fonctionnelle, des caractéristiques recherchées, etc.). Ceci implique souvent l'utilisation de généralisations et les problèmes qui en découlent: l'obtention des généralisations n'est pas forcément aisée, les connaissances sur lesquelles elles sont construites étant attachées à des cas particuliers et, inversement, une généralisation n'est pas forcément utilisable dans toutes les circonstances où elle semble applicable.

La conception innovatrice s'appuie sur trois méthodes:

- le raisonnement par analogie (case-based reasoning)
- la transformation structurelle
- la combinaison de sources de connaissances

4.6.1 Le raisonnement par analogie

Le raisonnement par analogie est la génération d'une solution en utilisant celle apportée à un problème antérieur. Le raisonnement par analogie implique plusieurs opérations [73, 47]:

- la *recherche des cas pertinents*: cette recherche se fait par raffinement des concepts, la base de cas étant organisée de manière hiérarchique. Les caractéristiques les plus générales sont recherchées en premier, puis on descend dans les niveaux d'abstraction.
- la *sélection du meilleur candidat*: chaque cas relevant est évalué et le meilleur est sélectionné. Les critères d'évaluation peuvent porter sur le nombre de caractéristiques similaires au problème posé, des heuristiques de préférence, une analyse dimensionnelle.
- la *construction de la solution*: cette construction se fait généralement par application d'opérateurs sur le candidat sélectionné, afin de l'adapter au nouveau problème.
- l'*évaluation de la solution*: la solution est analysée et évaluée selon les objectifs du problème posé.
- la *mise à jour de la base de cas*: la solution retenue est ajoutée à la base de cas pour son utilisation possible pour un problème futur.

L'efficacité de la méthode du raisonnement par analogie dépend de la taille de la base de cas et de la manière dont les cas sont stockés dans la base. L'information stockée contient non seulement la description physique des cas, mais aussi des données sur les fonctions réalisées, les comportements productibles et les performances. Afin de faciliter la construction d'une nouvelle solution, les opérateurs ayant servi à l'élaboration du cas sont aussi stockés.

Comme exemples de système utilisant le raisonnement par analogie, on peut citer:

- BOGART [67], un système de conception de circuits VLSI utilisant les résultats de VEXED. BOGART réutilise les solutions antérieures en réexécutant les plans de conception qui ont amené à ces solutions, et dont les prises de décisions sont adaptées aux nouvelles circonstances.
- ARGO [67], également un système de conception pour circuits VLSI. ARGO utilise l'expérience issue des solutions antérieures sous la forme de macro-règles. Les macro-règles sont obtenues à partir de graphes de règles de dépendance, représentant les plans de conception des solutions antérieures. Les macro-règles sont ensuite ordonnées dans un plan le plus spécifiquement applicable pour résoudre le nouveau problème.
- CADET [67], un système de conception pour mécanisme de contrôle de débit. Dans CADET, si il n'y a pas de solutions antérieures qui correspondent directement aux spécifications du nouveau problème, le système applique des transformations aux spécifications des comportements recherchés, jusqu'à ce qu'elles ressemblent à l'une des solutions antérieures (c'est-à-dire que la solution antérieure produise véritablement le comportement recherché ou quelque chose de similaire). Ces transformations peuvent s'appliquer à une partie des spécifications et faire ainsi correspondre plusieurs solutions antérieures qui sont alors regroupées dans la nouvelle solution.

4.6.2 La transformation structurelle

Une autre méthode de la conception innovatrice est la transformation structurelle. Cette méthode consiste à modifier une structure physique d'une certaine manière (guidée par des critères fonctionnels et de limitation de ressources), puis à analyser la fonctionnalité résultante. Les résultats de l'analyse permettent alors de suggérer de nouvelles modifications jusqu'à ce que la fonctionnalité désirée soit obtenue.

Un exemple de tel système est présenté dans [69]. Le problème consiste dans la simplification d'une solution modulaire (dans le sens que chaque élément de la structure physique implante une fonction différente) en exploitant les possibilités de partage fonctionnel (façon de regrouper dans un seul élément plusieurs fonctions). La transformation de la structure physique se fait par suppression d'un de ses éléments constitutifs. Une fois l'élément supprimé (et donc une fonction éliminée), le système recherche dans la structure restante les éléments qui peuvent être modifiés de manière à remplir la fonction disparue (tout en maintenant les fonctions que ceux-ci réalisent). Ces éléments identifiés sont alors modifiés afin de réaliser le mieux possible la fonction supprimée.

4.6.3 La combinaison de sources de connaissances

Pour certains problèmes de conception, plusieurs sources de connaissances peuvent être à disposition et chacune est potentiellement utile. Pour résoudre le problème, il est alors nécessaire de combiner ces différentes sources.

ALADIN [58], un système de conception pour des alliages d'aluminium, est basé sur ce principe. Une solution est obtenue par la combinaison de quatre sources d'expertise:

- une base de cas contenant les alliages déjà conçus et leurs propriétés
- une base de règles qui associe les modifications structurelles (par exemple: l'adjonction de magnésium à l'alliage) aux modifications fonctionnelles (par exemple: l'augmentation de la résistance de l'alliage)
- des modèles mathématiques des propriétés physiques
- des méthodes statistiques d'interpolation et d'extrapolation

Dans le même ordre d'idées, Meta-COOP [72] est un environnement de conception basé sur la programmation orientée-objet. Il permet l'intégration et l'utilisation de diverses représentations des connaissances et de différentes méthodes d'inférence, et peut travailler avec de l'information multi-média.

4.7 Les méthodes de la conception créatrice

La conception créatrice n'est l'objet d'études que depuis peu d'années. Les méthodes développées ne sont encore qu'à l'état d'ébauches, et cherchent à s'appuyer sur les méthodes de la conception innovatrice.

L. M. Wills et J. L. Kolodner [76] proposent une extension du raisonnement par analogie, en y ajoutant une étape de *redescription du problème*. Cette étape permet de déduire des relations entre le problème courant et des problèmes

similaires dans d'autres domaines, et de faciliter le transfert des idées entre contextes qui se chevauchent. La redescription permet également une redéfinition en termes plus concrets et plus détaillés au fur et à mesure du processus de conception.

Bien que les systèmes multi-agents, de par leur caractère dynamique, adaptatif et stochastique, paraissent constituer une approche intéressante de la conception créatrice, il semble que cette voie n'a pas encore été explorée.

4.8 La conception simultanée

La nécessité de s'adapter rapidement au marché et la réduction des cycles de vie des produits ont forcé les industriels à modifier leur processus de conception au sein de l'entreprise: alors qu'auparavant chaque département s'occupait de ses problèmes spécifiques (études de marché, élaboration du cahier des charges, conception du produit, production) avant de transmettre le dossier plus loin, ils cherchent actuellement à collaborer le plus tôt possible afin de minimiser les délais et les coûts.

Cette intégration des différents aspects de la conception d'un produit est appelée la *conception simultanée* ou encore ingénierie simultanée. Le but de cette approche est de maximiser le transfert d'information entre les différents experts, afin que les conséquences des décisions ou des impératifs puissent être prises en compte le plus tôt possible lors de la conception du produit. Cela évite de nombreux allers-retours entre bureau d'études et département des méthodes, et conduit souvent à des solutions plus simples et donc moins coûteuses.

Les outils d'intégration utilisés en conception simultanée offrent un langage commun entre les différents experts (humains ou outils de conception) et contrôlent les interactions entre ceux-ci:

- ParMan [46] est un outil s'appliquant aux problèmes de conception paramétriques. Il permet de définir et d'analyser les contraintes sur un ensemble de paramètres de sources différentes. Il maintient la cohérence d'un ensemble de contraintes distribuées en interagissant avec les experts, et donne à chaque participant une vue locale de cet ensemble.
- PACT [14] interconnecte des outils de conception au travers d'un langage de communication (comportant des opérations telles que requête, affirmation) dans un format d'échange d'information construit sur un vocabulaire commun. Chaque outil possède en entrée un programme qui coordonne la communication en sélectionnant le bon interlocuteur, en traduisant les messages à l'usage de l'outil, et en contrôlant les résultats de son travail.

4.9 Conclusion

La conception est la transformation d'un ensemble de fonctions en une structure physique réalisable telle qu'elle produise ces fonctions.

Une fonction, en tant que rôle que remplit un objet, est un ensemble de comportements recherchés, qui permettent de décrire abstraitement une manière de réaliser les fonctions, en caractérisant des objets fonctionnels (énergie, information, types d'objet) et leurs interactions.

Une structure physique est un ensemble de comportements productibles, qui permettent de décrire ce que réalise la structure physique, en caractérisant des objets physiques (composants de la structure) et leurs interactions.

Le processus de conception est un parcours à travers l'espace fonctionnel et l'espace physique de la conception:

- le traitement de l'espace fonctionnel consiste dans la détermination et la sélection des comportements recherchés qui réalisent les fonctions souhaitées.
- le traitement de l'espace physique consiste dans la détermination et la sélection des comportements productibles qui composent la structure physique.

La conception est alors possible par identification entre comportements recherchés et comportements productibles.

Chapitre 5

L'Assemblage

5.1 Introduction

L'assemblage est l'art d'appliquer un processus industriel à la réalisation des liaisons entre les pièces constitutives d'un produit. De ce fait, le domaine de l'assemblage recouvre l'ensemble des problèmes liés à la dernière étape de la production industrielle.

L'assemblage représente, selon les produits, 20 à 60 pour cent du coût de fabrication (jusqu'à 70 pour cent dans le cas d'une production entièrement manuelle) et environ 30 pour cent des investissements en moyens de production pour les industries. Ces quelques chiffres montrent que l'importance économique de ce domaine n'est pas négligeable, et que son optimisation peut être une source d'économie appréciable.

Les origines de ces coûts sont multiples et constituent des spécificités de l'assemblage auxquelles sont confrontés les concepteurs de systèmes d'assemblage:

- *difficulté d'automatiser l'assemblage*: l'automatisation de certaines opérations d'assemblage nécessite des solutions technologiques coûteuses. Cela provient souvent de choix faits à la conception du produit et inadaptés à l'assemblage automatique, ou de la transformation directe d'un processus manuel en un processus automatique sans reconsidérer l'ensemble du problème. Une solution à ce problème est l'ingénierie concurrente, où tous les services de l'entreprise (service commercial, bureau d'études, service des méthodes, fabrication) collaborent à la conception (ou à la reconception) du produit, afin de minimiser les conséquences économiques.
- *multiplicité des processus d'assemblage*: il y a, en général, plus d'une manière d'assembler un produit. Chacune d'elles implique des objets, des opérations et par conséquent des ressources différentes. La difficulté consiste alors à déterminer le processus d'assemblage qui amènera à un système dont les performances seront intéressantes et à moindre coût.
- *opérations constitutives et opérations logistiques*: un processus d'assemblage est constitué d'opérations nécessaires à l'assemblage du produit (opérations constitutives) et d'opérations nécessaires au bon fonctionnement du système (opérations logistiques: transport, test, etc.). Les opérations

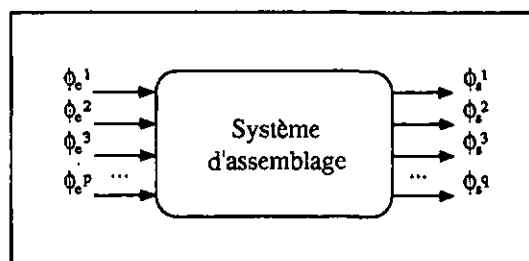


Figure 5.1: Représentation schématique d'un système d'assemblage

logistiques n'ajoutent aucune "valeur" utile au produit et sont consommatrices de temps. Comme elles impliquent des investissements en équipement comparables à ceux des opérations constitutives, il convient d'en minimiser le nombre par le choix judicieux du processus d'assemblage et de l'implantation géographique des équipements du système.

- *rendement des systèmes d'assemblage*: en production, un système d'assemblage effectue un grand nombre d'opérations différentes sur un grand nombre de composants différents. Ceci implique une fréquence d'incidents de toute nature (défaut d'alimentation, mauvaise exécution d'une opération, etc.) qui influencent la productivité et la qualité (et donc le coût de fabrication). La probabilité d'un incident dépend principalement des tolérances des composants impliqués et des imperfections des équipements incriminés. Afin de minimiser les conséquences des incidents les plus probables, il convient soit d'ajouter des opérations de détection et de récupération (ou des opérations de détection et des alarmes pour les cas les moins probables), soit de découpler par des stocks-tampons convenablement dimensionnés différents sous-ensembles de l'atelier.

Au vu de ces quelques points (parfois contradictoires), la conception de systèmes d'assemblage revient donc souvent à trouver un compromis entre technologie et coût, qui garantisse fiabilité et flexibilité afin d'obtenir un résultat rentable.

5.2 Définitions

Les définitions introduites ici sont celles données par C. Perrard [53]. Elles sont basées sur la notion de flux traversant un système d'assemblage.

Un système d'assemblage comporte en entrée p flux ϕ_e^i ($i = 1, \dots, p$), chacun constitué d'objets matériels c_i tous identiques et en sortie q flux ϕ_s^j ($j = 1, \dots, q$), chacun constitué d'objets matériels p_j tous identiques.

Les objets c_i sont appelés *composants élémentaires* et les objets p_j *produits finis*. Ces appellations sont relatives au système d'assemblage considéré, des composants élémentaires pouvant être des produits finis d'un système placé en amont, et des produits finis des composants élémentaires d'un système placé en aval.

Chaque produit fini p_j est le résultat de l'intégration de tout ou partie des composants élémentaires c_i selon une structure prédéfinie. La collection de com-

posants élémentaires c_i nécessaires à l'assemblage d'un exemplaire d'un objet p_j est appelé le *pré-produit* P_j . Les objets existants à un moment donné et issus des métamorphoses du pré-produit P_j afin d'obtenir le produit fini p_j , représentent des *produits intermédiaires*. Les objets particuliers, éléments d'un produit intermédiaire à un instant donné, sont appelés *constituants*; un *sous-assemblage* est un constituant qui n'est ni un composant élémentaire, ni un produit fini.

Les transformations appliquées à un constituant pour se rapprocher du produit fini sont des *opérations*. L'ensemble des opérations concernant la production d'un flux ϕ_j^s est appelé *processus d'assemblage*; une *gamme d'assemblage* est un processus d'assemblage où pour chaque opération, les moyens pour leur exécution ont été définis.

Un *système d'assemblage* est donc un ensemble d'*équipements* permettant la réalisation d'un ou plusieurs processus d'assemblage. Un système est appelé *système monoproduit* si le nombre de flux en sortie, q , est égal à 1 et *système multiproduit* si $q > 1$. Dans le cas multiproduit, on appelle *famille de produits* l'ensemble des q types d'objets produits en sortie.

5.3 Les processus d'assemblage

Un processus d'assemblage est un ensemble ordonné d'opérations qui amène des composants élémentaires au produit fini. Un processus d'assemblage ne fait pas référence à l'équipement, mais est une description fonctionnelle d'une manière d'assembler un produit. Un produit peut avoir plusieurs processus d'assemblage admissibles. La détermination des processus d'assemblage possibles se fait sur la base du produit, généralement par désassemblage du produit totalement assemblé et contrôle des contraintes, alors que leur sélection se fait sur des critères technologiques.

5.3.1 Analyse des processus d'assemblage

Les processus d'assemblage sont généralement exprimés à différents niveaux d'abstraction. La décomposition des tâches d'assemblage d'un niveau donné permet de préciser la description du processus d'assemblage, et d'analyser les implications éventuelles sur la sélection des équipements. Les niveaux de description présentés ci-après correspondent partiellement à la décomposition proposée par J. P. Bourrières [4].

Chaque niveau de description est caractérisé par un type d'opération et par les objets manipulés.

- **Niveau 0: assemblage et produit:** l'assemblage est la transformation du pré-produit en produit fini. A ce niveau de description (le plus grossier) n'apparaissent donc que ces deux objets et une seule opération: l'assemblage.
- **Niveau 1: tâches et constituants:** une *tâche d'assemblage* est une opération nécessaire à l'obtention du produit fini, c'est-à-dire qu'une tâche est la réalisation d'une caractéristique¹ donnée du produit. Une tâche d'assemblage est définie par la caractéristique elle-même, son (ses) constituant(s)

¹cf. 5.6.1

en entrée (avant réalisation) et son constituant en sortie (après réalisation). Le niveau 1 permet donc d'identifier les tâches formant le processus d'assemblage retenu pour réaliser l'assemblage du niveau 0 et les différents constituants générés par le processus.

- **Niveau 2: opérations et constituants localisés:** une *opération d'assemblage* est une transformation de l'état géométrique d'un constituant donné (position, orientation, nombre de pièces constitutives, degrés de liberté). Une opération d'assemblage est définie par la nature de la transformation et par les deux états géométriques, en entrée et en sortie, du constituant impliqué. Le niveau 2 permet donc d'identifier les opérations constitutives des tâches d'assemblages du niveau 1, et les états géométriques des constituants au cours du processus d'assemblage.
- **Niveau 3: fonctions élémentaires et constituants localisés:** une *fonction élémentaire* est une transformation élémentaire de l'état géométrique d'un constituant. La nature des fonctions élémentaires et celle des opérations d'assemblage sont similaires, c'est pourquoi le niveau 3 sert principalement à affiner la description des opérations du niveau 2.

5.3.2 Typologie des tâches d'un processus d'assemblage

Les tâches d'un processus d'assemblage portent sur l'appareil de production (personnes, machines, outils, etc.) et sur les objets (constituants) circulant dans le système.

Les **tâches de production**, concernant l'appareil de production uniquement (réglages, entretiens, changements d'outils, etc.), sont indispensables à l'évaluation du système d'assemblage retenu. Dans un premier temps, nous ne tiendrons pas compte des tâches de production, l'évaluation n'intervenant qu'en aval de notre travail.

Les **tâches d'assemblage**, concernant les objets uniquement, font seules partie de la typologie présentée ci-après. Cette typologie est la typologie proposée par J. M. Henrioud [40]. Les tâches d'assemblage sont subdivisées en deux catégories:

- **les tâches constitutives:** ce sont les tâches qui contribuent directement à la constitution du produit fini. Elles sont intrinsèques au produit fini et à son cahier des charges, et doivent être réalisées d'une manière ou d'une autre. Les tâches constitutives conditionnent la définition du processus d'assemblage.

Les tâches constitutives sont subdivisées en deux catégories selon leur nature binaire ou unaire:

- **les tâches relationnelles:** ce sont les tâches réalisant les caractéristiques binaires du produit. Elles sont décomposables en:
 - * **tâches géométriques:** une tâche géométrique est la réalisation d'un lien géométrique (contact) existant entre deux constituants du produit.
 - * **tâches physiques:** une tâche physique est la réalisation d'un (ou plusieurs) procédé(s) de solidarisation utilisé(s) pour bloquer

les liens géométriques: soudage, collage, rivetage, vissage, emmanchements, etc.

– les **tâches complémentaires**: ce sont les tâches de caractère unaire. Elles sont subdivisées en:

- * **tâches annexes**: une tâche annexe est la réalisation d'une caractéristique annexe du produit. Une caractéristique annexe ne contribue pas directement à l'assemblage du produit, mais elle doit être réalisée malgré tout durant le processus d'assemblage: peinture, marquage d'un constituant, usinage partiel, etc.
- * **tâches informationnelles**: une tâche informationnelle est une vérification de la bonne exécution des tâches constitutives: contrôle de qualité des différents sous-ensembles produits, réglage des mécanismes du produit, élimination des objets non conformes, etc.

• les **tâches logistiques**: ce sont les tâches qui portent sur les lieux de réalisation des tâches constitutives. Elles dépendent entièrement de l'architecture du système d'assemblage retenu.

Les tâches logistiques sont subdivisées en deux catégories:

- les **tâches positionnelles**: ce sont les transferts d'un lieu à un autre avec ou sans changement d'orientation.
- les **tâches de contrôle**: une tâche de contrôle est une vérification de la bonne exécution des tâches positionnelles: détection de présence, évaluation de la position ou de l'orientation des objets au sein du système de transfert, etc.

La figure 5.2 montre l'arbre de classification découlant de l'énumération faite ci-dessus.

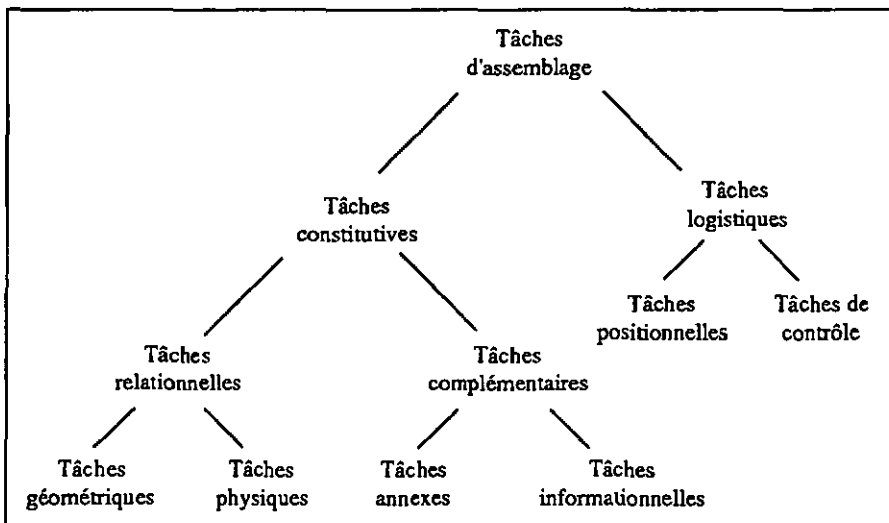


Figure 5.2: Typologie des tâches d'assemblage

5.4 Les systèmes d'assemblage

Le terme système d'assemblage est générique et il est utilisé pour désigner tout système qui comporte en entrée au moins un composant élémentaire et en sortie un produit fini. Ainsi, il s'applique aussi bien à une simple machine qu'à un atelier, pour peu qu'ils répondent à la définition précédente.

5.4.1 Structure des systèmes d'assemblage

La décomposition présentée ci-après est proposée par P. Olivier [51]. Elle fait apparaître les transports d'objets et les traitements qui leur sont appliqués, sans faire référence aux équipements. C'est une décomposition hiérarchique qui, à chaque niveau, fait apparaître des sous-systèmes eux-mêmes composés d'autres sous-systèmes de traitement. Ceux-ci sont considérés comme des boîtes noires interconnectées par des transports.

Ilot d'assemblage

L'*ilot d'assemblage* est un sous-ensemble des équipements d'un atelier alloué à l'assemblage complet d'un produit (ou d'une famille de produits).

La fonction de l'ilot d'assemblage est donc la transformation du pré-produit constitué de m composants élémentaires c_i en son produit fini p_j :

$$\text{ilot} : \{c_1, \dots, c_m\} \rightarrow p_j \quad (5.1)$$

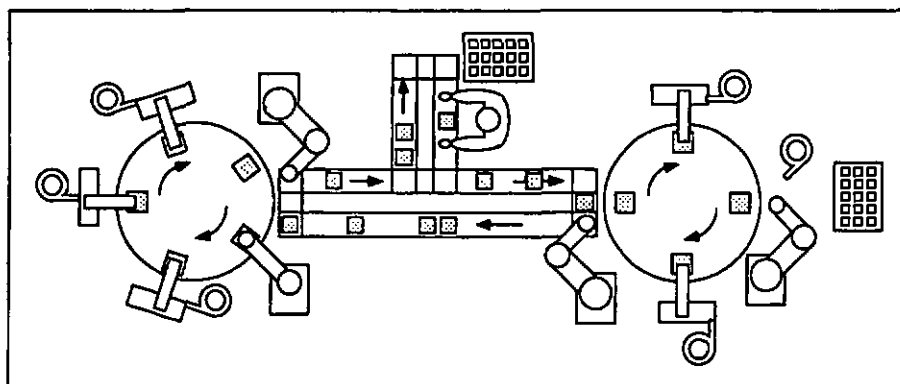


Figure 5.3: Exemple d'ilot d'assemblage constitué de trois cellules d'assemblage: les deux groupes de postes d'assemblage regroupés autour des tables tournantes et la cellule mono-poste où se trouve l'opérateur humain.

L'ilot d'assemblage est caractérisé par un horizon temporel significatif pour la gestion à ce niveau de description, et correspondant généralement à la durée des campagnes de production.

Les flux entrants dans un ilot d'assemblage correspondent essentiellement aux composants élémentaires fabriqués dans d'autres flots (usinage, moulage, etc.) du département de fabrication ou en provenance de l'extérieur (achat, sous-traitance, etc.). Selon les cas, il peut exister des flux entrants correspondant à de

l'outillage (palettes, chargeurs, préhenseurs interchangeables, etc.). Ces objets ne sont pas consommés par le processus d'assemblage, mais *recyclés*.

Les flux sortants d'un flot d'assemblage correspondent principalement aux produits finis. Les flux sortants secondaires correspondent aux objets recyclés, ainsi qu'aux produits défectueux.

Cellule d'assemblage

La *cellule d'assemblage* est sous-système d'un flot d'assemblage capable de produire de manière autonome et satisfaisante, durant une période limitée et non sécable en unités plus petites répondant à cette propriété. La cellule d'assemblage répond à des critères de fiabilité, l'expérience ayant montré qu'il n'est pas bon d'enchaîner rigidement plus de 6 à 12 opérations sans s'exposer à des incidents fréquents. Ainsi, l'flot d'assemblage est découpé en plusieurs groupes de machines (les cellules) reliés entre eux par des stocks-tampons: à court terme, ceux-ci permettent une certaine autonomie de production du groupe, même si l'un des groupes adjacents est arrêté.

La fonction de la cellule d'assemblage est la transformation de n ($n \leq m$) constituants c_i en un produit intermédiaire i déterminé:

$$\text{cellule} : \{c_1, \dots, c_n\} \rightarrow i \quad (5.2)$$

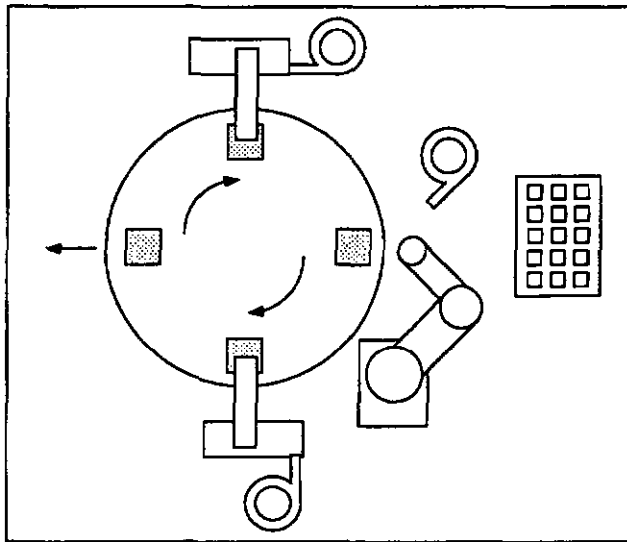


Figure 5.4: Exemple d'une cellule d'assemblage constituée de trois postes d'assemblage liés par une table tournante. Le quatrième poste de déchargement n'est pas représenté.

La cellule d'assemblage est caractérisée par:

- le *temps de cycle de la cellule* (t_{cc}): c'est la durée séparant l'apparition de deux produits intermédiaires i en sortie en fonctionnement normal.
- le *temps de traitement de la cellule* (t_{tc}): c'est la durée du séjour d'un exemplaire du produit à l'intérieur de la cellule.

- *l'en-cours de stockage*: c'est le nombre de produits intermédiaires i que peut produire la cellule dès l'instant où l'alimentation des stocks de la cellule est stoppée.

Les flux reliant une cellule d'assemblage à son environnement correspondent au renouvellement des stocks internes et à l'évacuation des produits intermédiaires. Ces flux peuvent être soit unitaires (les constituants concernés sont transportés individuellement), soit collectifs (les constituants sont regroupés et transportés en *paquets* à intervalle régulier).

Poste d'assemblage

Le *poste d'assemblage* est un sous-système d'une cellule d'assemblage produisant un produit intermédiaire donné en un temps donné, durant lequel un seul exemplaire du produit en cours de fabrication est présent dans le poste.

La fonction du poste d'assemblage est donc la transformation de o ($o \leq n \leq m$) constituants c_i en un produit intermédiaire i' déterminé:

$$\text{poste} : \{c_1, \dots, c_o\} \rightarrow i' \quad (5.3)$$

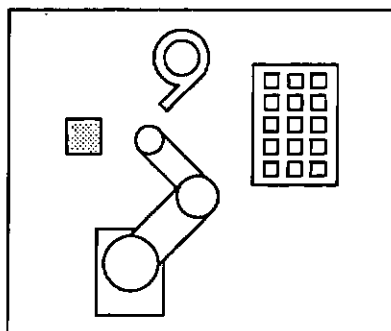


Figure 5.5: Exemple d'un poste d'assemblage constitué d'un posage, d'un bol vibrant, d'un plateau alvéolé et d'un robot.

Le poste d'assemblage est caractérisé par son *temps de cycle* (t_{cp}). Au sein de la cellule, les postes sont disposés les uns à la suite des autres, et reliés entre eux par des transports. Chacun des postes de la cellule travaille simultanément sur un exemplaire du produit.

J. Figour [26] définit 4 niveaux d'entrée pour un poste d'assemblage, chaque niveau étant caractérisé par les états géométriques des constituants impliqués:

- **Niveau N0**: c'est le lieu de la transformation effective des constituants en produit intermédiaire. Dans l'exemple de la figure 5.5, le niveau N0 est constitué du posage. Au niveau N0 correspond une tâche constitutive.
- **Niveau N1**: c'est le lieu où un constituant est isolé, et où sa position et son orientation sont connues, de manière à pouvoir être saisi. Le niveau N1 correspond à l'entrée du système de chargement. Dans l'exemple, une alvéole du plateau ou le séparateur en sortie du rail du bol vibrant sont au niveau N1. Entre le niveau N1 et le niveau N0, on trouve une tâche

positionnelle, amenant le constituant du niveau N1 au niveau N0 (avec ou sans orientation).

- **Niveau N2:** c'est l'entrée du système assurant l'autonomie de fonctionnement du poste à partir d'un ensemble de constituants reconnus et orientés. Le plateau alvéolé et l'entrée du rail du bol vibrant sont au niveau N2. Entre le niveau N2 et le niveau N1, on trouve une tâche positionnelle qui va amener les constituants du niveau N2 au niveau N1 en maintenant leur orientation et isoler l'un d'entre eux.
- **Niveau N3:** c'est l'entrée du système assurant la distribution à partir du vrac. Le vrac du bol vibrant est au niveau N3. Entre le niveau N3 et le niveau N2, on trouve une tâche positionnelle d'orientation.

Opération

L'*opération* est une transformation subie par un exemplaire de produit intermédiaire. Au sein d'un poste, le produit intermédiaire peut subir une ou plusieurs opérations, successivement ou simultanément.

La fonction d'une opération est donc la transformation de p ($p \leq o \leq n \leq m$) constituants c_i en un produit intermédiaire i'' déterminé:

$$\text{opération} : \{c_1, \dots, c_p\} \rightarrow i'' \quad (5.4)$$

On parle d'*opération unaire* lorsque $p = 1$ et d'*opération binaire* lorsque $p = 2$. Il est rare de rencontrer des opérations d'ordre supérieur ($p > 2$). L'opération est caractérisée par une durée t_{co} nécessairement inférieure ou égale au temps de cycle du poste t_{cp} .

Il peut exister des transports entre deux opérations d'un même poste si le constituant impliqué nécessite un déplacement court (réorientation par exemple) pour satisfaire aux conditions spatiales de l'opération aval.

5.4.2 Les équipements constitutifs des systèmes d'assemblage

Le paragraphe précédent donne une description de l'organisation des systèmes d'assemblage, sans faire référence aux moyens effectivement mis en oeuvre pour réaliser les différentes opérations du processus d'assemblage. Ce paragraphe présente une typologie des équipements à disposition pour constituer un système d'assemblage. L'être humain n'apparaît pas dans la typologie, sa flexibilité, son pouvoir d'adaptation et son intelligence le plaçant à part des machines. La typologie présentée est basée sur les travaux de J.-C. Prélaz [55].

Les équipements sont subdivisés en trois catégories: les *effecteurs*, l'*outillage* et les *éléments de structure*.

Les effecteurs

Les effecteurs sont les équipements chargés des mouvements de moyennes et longues envergures (relativement à la taille des objets manipulés). En général, ils n'agissent pas directement sur les objets, mais servent de porteurs pour l'outillage. De par le fait qu'ils sont relativement indépendants de l'application, on trouve sur le marché certains modèles standard.

Les effecteurs sont subdivisés en deux catégories:

- **les opérateurs:** les opérateurs sont les équipements qui permettent une orientation et un positionnement précis d'un objet (constituant ou outillage) entre deux points de l'espace. Ils sont donc commandés et offrent une certaine flexibilité d'adaptation à de nouvelles tâches. Dans cette catégorie entrent: les robots, les manipulateurs hydrauliques et pneumatiques, les manipulateurs numériques, les tables X-Y, etc.
- **les transferts:** les transferts sont les équipements de liaison entre les différents sous-systèmes d'un flot d'assemblage. Le transport des objets d'un sous-système à l'autre peut se faire de manière synchrone ou asynchrone, individuellement ou en paquets. Dans cette catégorie entrent: les convoyeurs incrémentaux et libres, les tables tournantes, les AGV, etc.

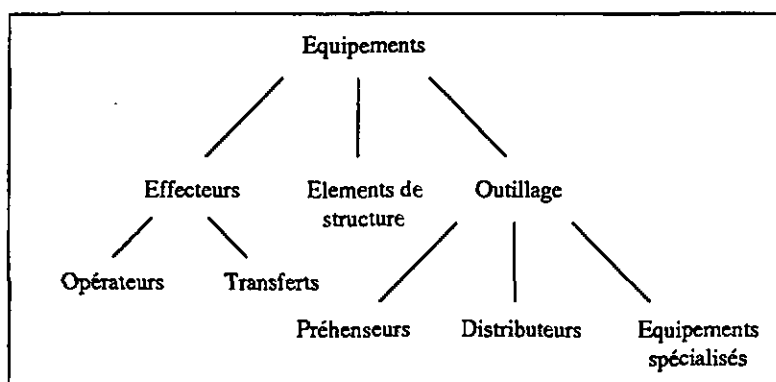


Figure 5.6: Typologie des équipements constitutifs des systèmes d'assemblage

L'outillage

L'outillage regroupe les équipements chargés de la manipulation directe des objets et des opérations particulières. Par définition, l'outillage dépend de l'application et nécessite donc un travail de conception particulier à chaque fois.

L'outillage est subdivisé en trois catégories:

- **les préhenseurs:** les préhenseurs sont les équipements de manipulation des constituants. Ils permettent leur référencement dans l'espace pour la réalisation des opérations. Dans cette catégorie entrent: les pinces mécaniques et à vacuum, les posages, les palettes, etc.
- **les distributeurs:** les distributeurs sont les équipements chargés de fournir au système d'assemblage des composants élémentaires isolés et référencés. Dans cette catégorie entrent: les bols et les rails vibrants, les chargeurs à bande, les plateaux alvéolés, etc.
- **les équipements spécialisés:** les équipements spécialisés regroupent les équipements dédiés à la réalisation d'opérations dépendantes d'un procédé de fabrication particulier. Dans cette catégorie entrent: les dispenseurs de colle, les postes à souder, les presses, les fours, etc.

Les éléments de structure

Les éléments de structure (bâti, supports, barrières, carénages, etc.), bien que n'agissant pas sur les objets, sont nécessaires au bon fonctionnement du système d'assemblage. Ils constituent l'ossature du système, et ont aussi un rôle de sécurité en permettant le confinement des zones dangereuses.

5.5 Processus de conception des systèmes d'assemblage

Le processus de conception pour les systèmes d'assemblage présenté ci-après est une compilation des méthodes proposées par le Laboratoire d'Automatique de Besançon [3, 53], par J. Figour [26] et par le Charles Stark Draper Laboratories [75].

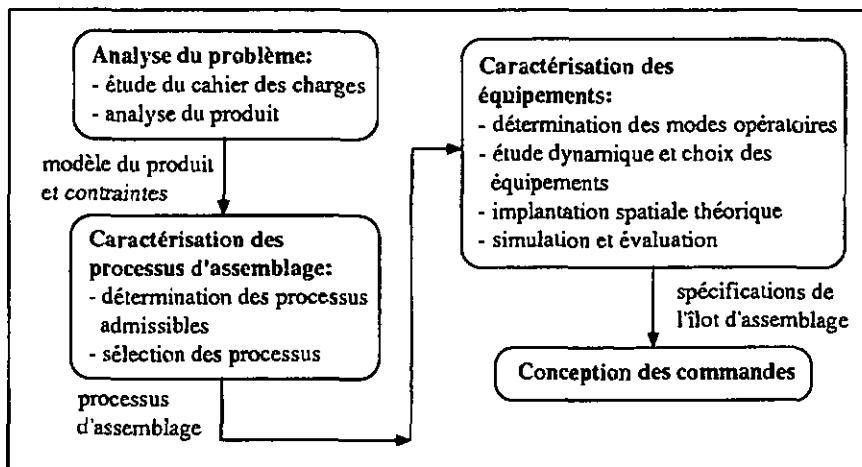


Figure 5.7: Processus de conception des systèmes d'assemblage

Le processus de conception est constitué de quatre étapes (cf. figure 5.7):

- *l'analyse du problème*: l'étude du cahier des charges et l'analyse du produit à assembler permettent de mettre en évidence les données et les contraintes du problème. Le résultat principal de cette étape est un modèle du produit et des contraintes associées.
- *la caractérisation des processus d'assemblage*: cette étape détermine les processus d'assemblage admissibles pour le produit, et sélectionne ceux qui favorisent au mieux certaines stratégies. Cette étape constitue en fait la construction et le parcours de l'espace fonctionnel² de la conception des systèmes d'assemblage.
- *la caractérisation des équipements*: cette étape détermine et sélectionne les équipements constitutifs et leur organisation au sein du système d'assemblage, et fait une évaluation du comportement dynamique du système

²cf. section 4.4.1

conçu. Cette étape constitue en fait la construction et le parcours de l'espace physique ³ de la conception des systèmes d'assemblage.

- la *conception des commandes*: la conception du système d'assemblage est complétée par l'élaboration des différents niveaux de commande qui réguleront le travail du système d'assemblage.

Les sections suivantes présentent plus en détail chacune de ces étapes et les différentes solutions qui leur ont été apportées.

5.6 L'analyse du problème

En amont, le concepteur de systèmes d'assemblage dispose de deux sources d'information: le cahier des charges et le produit à assembler.

Le cahier des charges fixe les objectifs de la production (en termes de quantité à produire) et les contraintes pouvant avoir trait par exemple aux équipements (obligation de réutiliser des équipements existants), aux modes de gestion (flux poussé ou flux tiré), à la qualité (imposition de procédures de test), au degré d'automatisation, aux investissements autorisés, etc. L'étude du cahier des charges permet l'identification de ces contraintes.

L'analyse du produit est basée sur le dessin d'ensemble du produit, les dessins de détail des composants et sur des échantillons. Les principaux résultats de l'analyse du produit sont:

- l'identification de l'ensemble des tâches constitutives qui formeront les processus d'assemblage
- une modélisation du produit fini

5.6.1 Les modélisations du produit

Les modélisations que l'on peut trouver dans la littérature peuvent être classées en deux catégories complémentaires: les *modélisations fonctionnelles* et leur extension, les *modélisations opératoires*.

Modélisations fonctionnelles

Les modélisations fonctionnelles s'appuient sur le concept de *liaison fonctionnelle*. Une liaison géométrique (contact entre deux pièces) est rendue fonctionnelle (dans le sens qu'elle réalise la fonction souhaitée) en lui appliquant un procédé de solidarisation [26]. Par exemple: le contact entre un composant SMD et le circuit imprimé ne pourra conduire le courant (fonction) qu'une fois la soudure effectuée.

Une modélisation fonctionnelle est une modélisation *orientée produit*, dans le sens qu'elle représente ce qu'est le produit du point de vue de l'assemblage.

Pour A. Bourjault [3], il existe une et une seule liaison fonctionnelle entre deux composants élémentaires c_e^i et c_e^j d'un produit p^k donné, si il existe au moins une liaison mécanique entre ces deux composants. Les liaisons fonctionnelles sont notées l_f^1, \dots, l_f^n . A tout produit p^k est associé un graphe non orienté

³cf. section 4.4.2

$G = \langle S, A \rangle$, appelé *graphe des liaisons fonctionnelles* (cf. figure 5.8), tel que l'ensemble S de ses sommets corresponde de façon bi-univoque à l'ensemble $C_e^{p^k} = \{c_e^1, \dots, c_e^m\}$ des composants élémentaires de p^k , et l'ensemble A de ses arêtes à l'ensemble $L_f^{p^k} = \{l_f^1, \dots, l_f^n\}$ des liaisons fonctionnelles de p^k . G est un graphe simple (il y a au plus une liaison fonctionnelle entre deux composants élémentaires c_e^i et c_e^j et il n'y a pas de liaison fonctionnelle entre un composant élémentaire c_e^i et lui-même) et connexe. Par simplification, les composants de solidarisation (écrous, vis, rivets, etc.) ne sont pas considérés comme des composants élémentaires $\in C_e^{p^k}$, mais comme appartenant aux procédés de solidarisation.

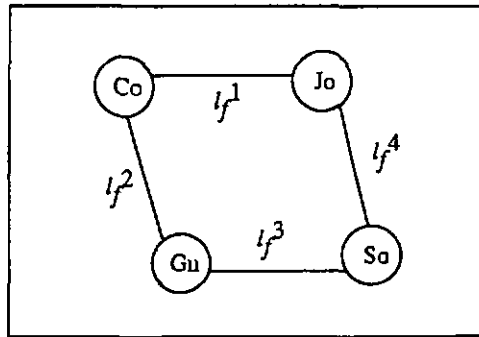


Figure 5.8: Graphe des liaisons fonctionnelles de la valve de vélo, selon la modélisation de A. Bourjault

Pour C. Mascle [48], une liaison fonctionnelle est égale à une liaison de prépositionnement en conjonction ou non avec une liaison d'attachement. Les liaisons sont décrites par des matrices qui expriment les demi-degrés de liaison. Les demi-degrés de liaison correspondent aux translations et rotations relatives entre deux composants élémentaires selon les demi-axes d'un repère orthonormé. A tout produit p^k est associé un graphe non orienté $G = \langle S, A \rangle$ (cf. figure 5.9) tel que l'ensemble S de ses sommets corresponde de façon bi-univoque à l'ensemble $C_e^{p^k}$ des composants élémentaires de p^k , constitué de C_{ef} , l'ensemble des composants fonctionnels et de C_{es} , l'ensemble des composants de solidarisation. L'ensemble A des arêtes de G correspond à l'ensemble $L_f^{p^k}$ des liaisons fonctionnelles de p^k .

Le modèle proposé par L. S. Homem de Mello et A. C. Sanderson [42] distingue les pièces, les contacts et les attachements. Il comprend aussi les relations entre ces entités. Le produit est décrit par le quintuplet $\langle P, C, A, R, a - functions \rangle$ avec:

- P , l'ensemble des symboles identifiant chacun une pièce
- C , l'ensemble des symboles identifiant chacun un contact entre deux pièces.
- A , l'ensemble des symboles identifiant chacun un attachement qui agit sur un ensemble de contact.
- R , l'ensemble des symboles identifiant chacun une relation au sein d'une paire d'éléments de $P \cup C \cup A$

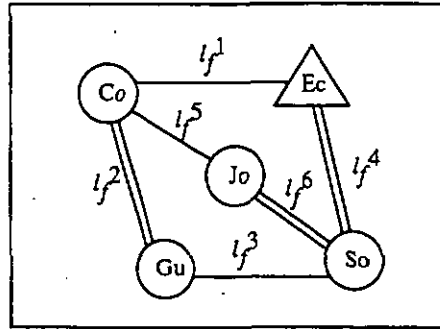


Figure 5.9: Graphe des liaisons de la valve de vélo, selon la modélisation de C. Mascle. Un arc simple représente une liaison de prépositionnement, et un arc double, une liaison de prépositionnement avec une liaison d'attachement. Les composants de solidarisation sont représentés par un triangle.

- a – *functions*, l'ensemble des fonctions d'attributs dont les domaines sont des sous-ensembles de $P \cup C \cup A \cup R$. Ces fonctions associent aux entités ou relations, leurs caractéristiques (par exemple: la forme d'une pièce, le type d'attachement, etc.)

Ce modèle, appelé *modèle relationnel*, peut être représenté par un graphe non orienté $G = \langle S, A \rangle$ (cf. figure 5.10) tel que l'ensemble S de ses sommets corresponde de façon bi-univoque à l'ensemble $P \cup C \cup A$. L'ensemble A des arêtes de G correspond à l'ensemble R des relations entre les entités du produit.

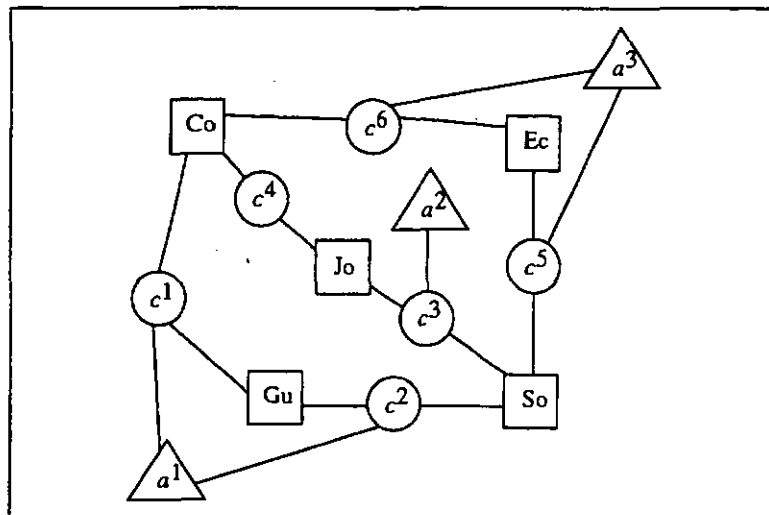


Figure 5.10: Modèle relationnel de la valve de vélo, selon la modélisation de L. S. Homem de Mello et A. C. Sanderson. Les contacts $c^i \in C$ sont représentés par des cercles et les attachements $a^i \in A$ par des triangles.

Modélisations opératoires

La modélisation opératoire est basée sur l'ensemble des caractéristiques du produit fini qui doivent être réalisées au cours de tout processus retenu pour son assemblage. Les caractéristiques du produit à réaliser sont déterminées au cours de la phase initiale d'analyse du produit. Elles sont de trois types:

- les *caractéristiques géométriques*: ce sont les liens géométriques (contacts) existants entre les pièces du produit
- les *caractéristiques physiques*: ce sont les procédés de solidarisation utilisés pour bloquer les liens géométriques: soudage, collage, rivetage, vissage, emmanchements, etc.
- les *caractéristiques complémentaires*: ce sont les caractéristiques qui ne sont pas à proprement parler des caractéristiques d'assemblage, mais qui doivent être réalisées malgré tout durant le processus d'assemblage: peinture, impression, test, usinage partiel, etc.

Une modélisation opératoire est une modélisation *orientée processus*, dans le sens qu'elle représente ce qui doit être fait durant le processus d'assemblage.

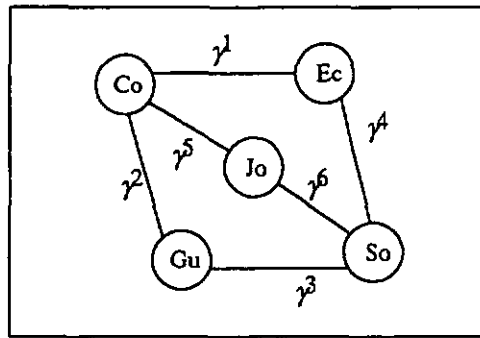


Figure 5.11: Graphe du modèle opératoire de la valve de vélo, selon la modélisation de J. M. Henrioud.

Le modèle opératoire de J. M. Henrioud [40] prend en compte séparément la structure du produit (composants élémentaires et caractéristiques géométriques) et les caractéristiques physiques et complémentaires. La structure du produit p^k est modélisée par un graphe $G = \langle S, A \rangle$ (cf. figure 5.11) tel que l'ensemble S de ses sommets corresponde à l'ensemble $C_e^{p^k} = \{c_e^1, \dots, c_e^m\}$ des composants élémentaires de p^k , et l'ensemble A de ses arêtes à l'ensemble Γ^{p^k} des caractéristiques géométriques de p^k . Les caractéristiques physiques et complémentaires sont modélisées par deux applications f et g respectivement:

- l'application f fait correspondre à toute caractéristique physique $\in \Phi^{p^k}$, l'ensemble des caractéristiques physiques de p^k ; l'ensemble des constituants $\in \Pi(C_e^{p^k})$, l'ensemble des parties de $C_e^{p^k}$, entre lesquels cette caractéristique doit être établie:

$$f : \Phi^{p^k} \rightarrow \Pi(C_e^{p^k}) \quad (5.5)$$

- l'application g fait correspondre à toute caractéristique complémentaire $\in \Delta^{p^k}$, l'ensemble des caractéristiques complémentaires de p^k , le sous-ensemble de p^k concerné par cette caractéristique. Un sous-ensemble est caractérisé par ses constituants $\in \Pi(C_e^{p^k})$, les liens géométriques devant nécessairement exister entre ces constituants et éventuellement les caractéristiques physiques ou complémentaires déjà réalisées $\in \Pi(\Gamma^{p^k} \times (\Phi^{p^k} \cup \Delta^{p^k}))$

$$g : \Delta^{p^k} \rightarrow \Pi(C_e^{p^k}) \times \Pi(\Gamma^{p^k} \times (\Phi^{p^k} \cup \Delta^{p^k})) \quad (5.6)$$

5.7 Caractérisation des processus d'assemblage

Cette étape comporte deux parties: la détermination des processus d'assemblage admissibles et la sélection du(des) processus le(s) "meilleur(s)".

5.7.1 Détermination des processus admissibles

Un processus d'assemblage est admissible lorsque toutes ses opérations constitutives sont réalisables dans le contexte que leur fournit le processus d'assemblage. Une opération constitutive est réalisable dans un contexte si elle répond à certaines contraintes. Ces contraintes sont appelées *contraintes-produit* car elles tirent leur origine dans le produit lui-même. Elles sont les conditions à remplir pour la réalisation de l'opération:

- **la géométrie des constituants:** la géométrie de base des constituants (forme, taille, tolérances, etc.) impliqués dans une opération doit offrir des surfaces appropriées pour leur préhension. De plus, ces surfaces doivent être facilement accessibles aux équipements.
- **l'accessibilité:** pour mettre en contact deux constituants, il doit exister une trajectoire sans collision qui les amène en contact.
- **la stabilité:** les deux constituants qui doivent être assemblés doivent avoir des orientations telles que leurs pièces constitutives maintiennent leurs positions relatives, et que les contacts ne soient pas rompus spontanément.

La génération des processus d'assemblage se fait par décomposition du modèle du produit fini et contrôle des contraintes:

- A. Bourjault [3], dans son système SAGA, considère le résultat de la satisfaction des contraintes-produit comme des contraintes de précedence pour l'établissement des liaisons fonctionnelles. Le système pose des questions à un expert humain sur les contraintes de précedence, qui sont ensuite utilisées pour décomposer le graphe des liaisons fonctionnelles. Le CSDL [75] utilise la même approche, mais en réduisant l'ensemble des questions posées à l'expert.
- J. M. Henrioud [40, 41] utilise dans son système LEGA la même approche que L. Homem de Mello [42]. Le modèle opératoire est décomposé de manière itérative en commençant par le produit fini: à chaque itération,

les opérations géométriques (et non-géométriques) possibles sont générées par les décompositions possibles du graphe. Le système contrôle alors leur faisabilité au moyen d'une base de faits regroupant les contraintes. La base de faits est augmentée au fur et à mesure de la décomposition, soit au moyen de règles, soit par des questions à l'expert humain.

- J. Figour [26], repris par C. Masclé [48], propose une méthode de décomposition basée sur l'analyse du désassemblage du produit totalement assemblé. A chaque étape, les composants pouvant être désassemblés (en commençant par les composants d'attachement) sont déterminés en contrôlant les accessibilités et la stabilité des sous-assemblages obtenus.

Nous citerons encore R. H. Wilson et J.-C. Latombe [77], qui génèrent les processus d'assemblage par une analyse géométrique des trajectoires de désassemblage du produit. Cette analyse permet également l'évaluation et la sélection des processus générés.

5.7.2 Sélection des processus

Certains processus d'assemblage favorisent mieux certaines stratégies d'assemblage que d'autres. Ces stratégies peuvent être représentées sous la forme de critères de sélection qui sont par exemple [53]:

- des critères relatifs aux sous-assemblages: certains sous-assemblages doivent obligatoirement apparaître au cours du processus d'assemblage (contrôle de qualité et de fonctionnalité du produit, fabrication de sous-ensembles de rechange pour le service après-vente, etc.)
- des critères relatifs aux composants: les opérations portant sur des composants ayant des caractéristiques identiques sont regroupées dans le processus d'assemblage (morphologie, dimensions, poids, fragilité, etc.)
- des critères relatifs aux liaisons entre composants: les opérations ayant des caractéristiques identiques sont regroupées dans le processus d'assemblage (mise en oeuvre d'un même procédé de solidarisation, directions d'assemblage identiques, etc.)
- des critères relatifs au processus d'assemblage: l'architecture du système d'assemblage recherché est fortement influencé par le taux de parallélisme des différentes opérations qui constituent le processus d'assemblage.

Le choix de l'une ou l'autre stratégie est généralement fixé par le cahier des charges ou dépend des préférences du concepteur.

5.7.3 Représentation des processus d'assemblage

Il existe plusieurs manières de représenter les processus d'assemblage, chacune présentant des avantages et des inconvénients. Le choix de l'une ou l'autre dépend de la méthode qui les exploite et du problème à résoudre. Ces représentations sont exposées et analysées ci-après.

Les graphes de précédence

Un graphe de précédence est un graphe orienté dont les noeuds représentent des événements ou des éléments caractéristiques du phénomène à représenter, et où chaque arc introduit une contrainte d'antériorité temporelle ("doit être réalisé avant") entre la réalisation des deux événements ou entre les deux éléments représentés par les deux sommets que lie cet arc.

Pour représenter des processus d'assemblage, les noeuds du graphe de précédence représentent soit les composants élémentaires, soit les tâches du processus d'assemblage. En général, un graphe de précédence permet de représenter un ensemble de processus d'assemblage avec l'avantage de présenter l'information sous une forme compacte.

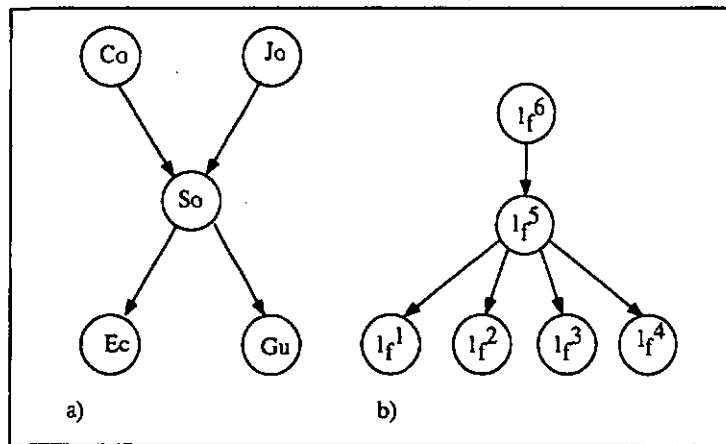


Figure 5.12: Exemples de graphes de précédence pour la valve de vélo: ils représentent les processus admissibles. a) graphe basé sur les composants élémentaires. b) graphe basé sur les liaisons fonctionnelles telles que décrites dans la figure 5.9.

Par contre, les graphes de précédence présentent les inconvénients suivants [53]:

- les contraintes prises en compte pour l'élaboration du graphe de précédence ne concernent en général que des tâches mettant en oeuvre un composant élémentaire ou un sous-assemblage. Les contraintes plus complexes (mettant en cause deux sous-assemblages) risquent de ne pas être représentées, avec la conséquence de perdre certains processus d'assemblage ou de produire des processus faux.
- la complexité de réalisation des tâches d'assemblage dépend en partie du contexte (états géométriques des constituants impliqués). Les graphes de précédence ne peuvent pas faire apparaître le contexte, ce qui signifie que la comparaison entre tâches de deux processus particuliers ne peut se faire.
- les contraintes de précédence traduisent des exigences de natures différentes: contrainte-produit, contrainte-ressources, préférence de l'ingénieur, etc. La non hiérarchisation de leur prise en compte peut amener à

des solutions peu intéressantes et à la non exploration de l'ensemble des solutions admissibles.

Les graphes d'états

D. E. Whitney et al. [75] ont proposé de représenter l'ensemble des processus d'assemblage d'un produit par un seul graphe: le graphe d'états.

Un graphe d'états est un graphe dont:

- les noeuds représentent les états possibles du produit, codés par des mots binaires $E = L_1, L_2, \dots, L_n$ avec L_i la i -ème liaison fonctionnelle du produit. Il y a bijection entre les noeuds du graphe et l'ensemble de tous les états possibles des produits intermédiaires.
- les arcs représentent les tâches permettant de passer de l'état E_k à l'état E_{k+1} du produit intermédiaire. Les arcs sont orientés de manière à respecter l'ordre chronologique d'apparition des états du produit intermédiaire. Il n'existe qu'une et une seule tâche pour passer de l'état E_k à l'état E_{k+1} .

Les graphes d'états sont des graphes orientés, simples (un seul arc entre deux noeuds et sans boucle) et connexes. Ils possèdent un seul noeud sans prédécesseur (représentant le pré-produit $E = 0, 0, \dots, 0$) et un seul noeud sans successeur (représentant le produit fini $E = 1, 1, \dots, 1$)

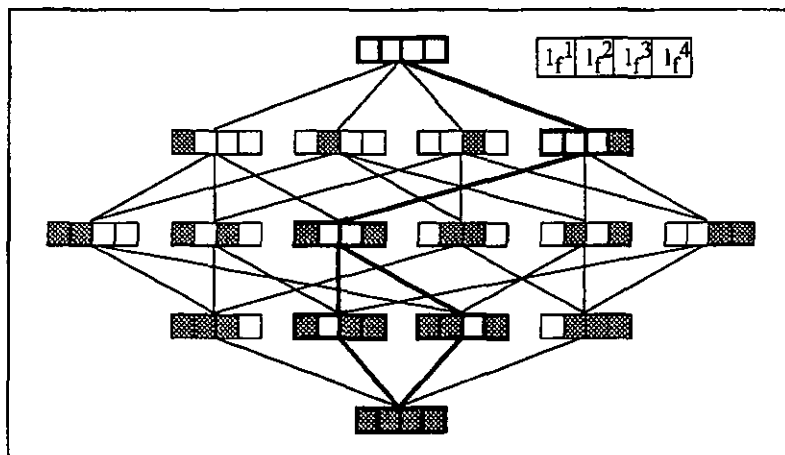


Figure 5.13: Graphe d'états de la valve de vélo basé sur liaisons fonctionnelles telles que décrites dans la figure 5.8. Seuls les processus en gras sont des processus admissibles.

Ce type de représentation offre l'avantage d'être compact, mais comme pour les graphes de précedence, le contexte d'exécution des tâches n'apparaît pas, puisque les sous-assemblages ne sont pas représentés.

Les arbres d'assemblage

Un arbre d'assemblage est un graphe arborescent dont la racine est le produit fini, les noeuds sont les sous-ensembles du produit fini et les feuilles, les com-

posants élémentaires. Chaque sous-ensemble situé à un noeud non-terminal est réalisé à partir de ses k successeurs immédiats:

- par une tâche géométrique si $k \geq 2$
- par une tâche non-géométrique si $k = 1$

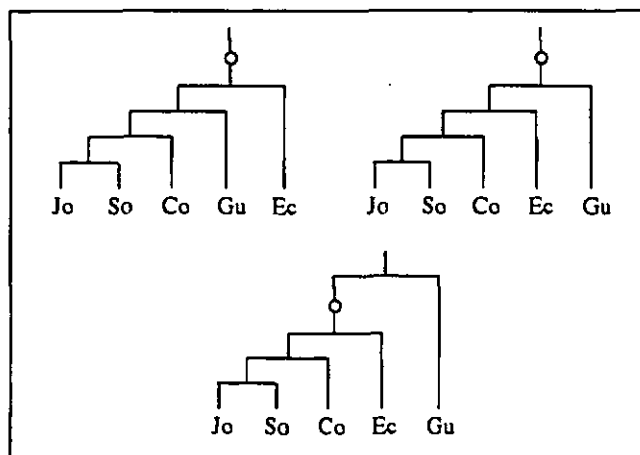


Figure 5.14: Trois arbres d'assemblage pour la valve de vélo. Le cercle est le noeud où apparaît le sertissage de la soupape.

Un arbre d'assemblage ne permet la représentation que d'un seul processus d'assemblage.

Les graphes ET/OU

Un graphe ET/OU [59, 43] est un hypergraphe dont les noeuds représentent les divers constituants générés au cours des processus d'assemblage (chaque noeud représentant un constituant est différent, et chaque constituant est représenté) et les arcs, les tâches réalisant les constituants. Chaque tâche met en oeuvre plusieurs constituants au départ (condition ET, représentée par des hyperarcs) et un constituant peut être le résultat de plusieurs tâches (condition OU, représentée par les différents arcs arrivant sur le noeud).

Un graphe ET/OU est une représentation collective d'arbres d'assemblage qui offre l'avantage d'être très compacte, malgré le fait que les opérations ne sont pas mises en évidence.

Les réseaux de Petri

Un réseau de Petri peut représenter un ensemble de processus d'assemblage:

- les places du réseau constituent l'ensemble des constituants générés au cours des processus d'assemblage (comme dans le graphe ET/OU)
- les transitions du réseau constituent l'ensemble des opérations

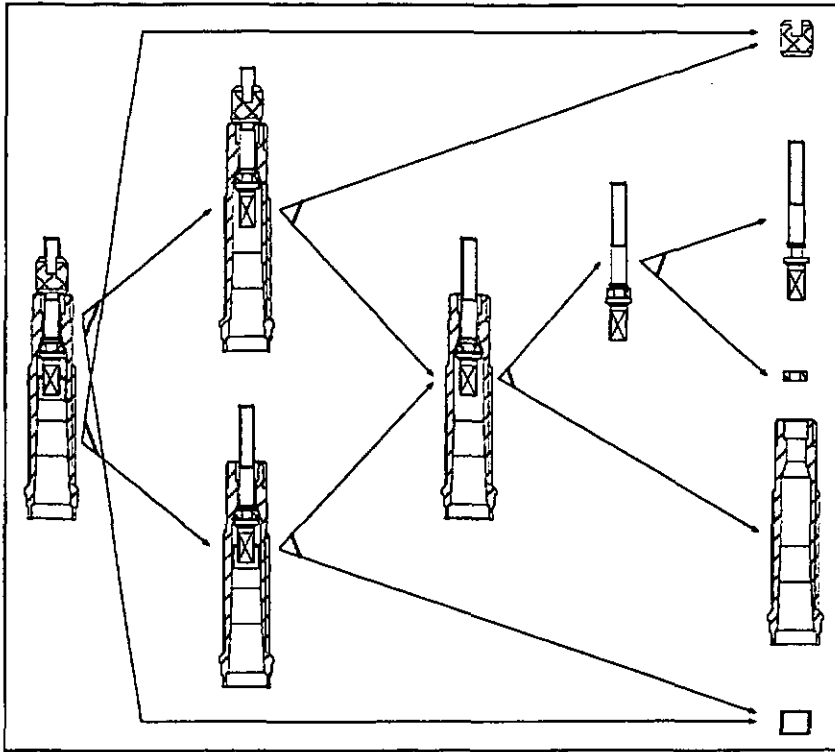


Figure 5.15: Graphe ET/OU de la valve de vélo représentant les processus admissibles. Il est construit sur le modèle de la figure 5.10.

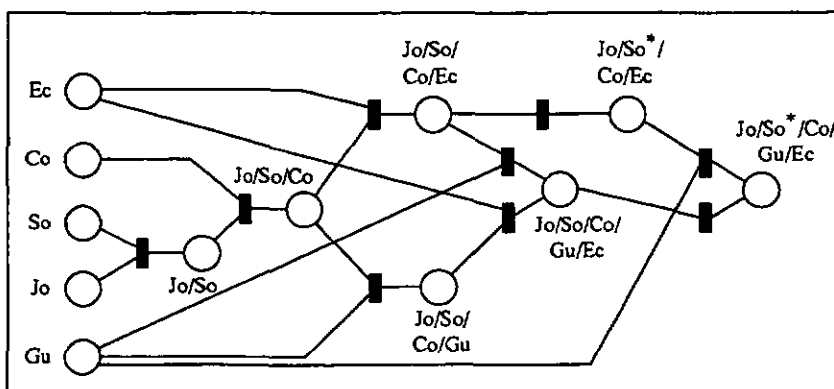


Figure 5.16: Réseau de Petri représentant les processus admissibles de la valve de vélo. So^* indique que la soupape est sertie.

- les jetons du réseau représentent les constituants réels, et le marquage des places du réseau par ces jetons donne instantanément l'état du produit intermédiaire.

Outre l'avantage d'être compacte, une représentation par réseau de Petri permet de représenter le contexte de chaque opération. Par contre, il peut apparaître des problèmes de "dead-lock" par la fabrication intempestive de sous-assemblages ne permettant pas de réaliser le produit fini. C'est pourquoi leur utilisation doit être faite avec prudence.

De par leur aspect dynamique, les réseaux de Petri, pour représenter des processus d'assemblage, sont surtout utilisés pour la simulation ou pour le suivi de la production.

5.8 La caractérisation des équipements

La caractérisation des équipements implique un certain nombre de contraintes à respecter ou à résoudre. Ces contraintes, appelées *contraintes-ressources*, sont liées à la sélection et à l'intégration des équipements au sein de l'îlot d'assemblage. Elles sont principalement technologiques, temporelles et économiques:

- **la flexibilité:** c'est la capacité d'un équipement à traiter les variantes d'un produit ou d'effectuer différentes opérations. Le choix d'un équipement peut dépendre de sa flexibilité en réponse à la flexibilité attendue de la part de l'îlot d'assemblage, et définie dans le cahier des charges.
- **la fiabilité:** c'est l'efficacité d'un équipement. Elle peut être interprétée comme le rapport entre ce que peut faire l'équipement et ce qu'il a à réaliser. Elle dépend de la qualité de construction de l'équipement et des tolérances des objets manipulés. De la fiabilité des équipements dépend en partie le rendement de l'îlot d'assemblage.
- **l'ergonomie du layout:** l'intégration des différents équipements au sein de l'îlot d'assemblage doit permettre une intervention en cas de pannes ou de réglages, tout en minimisant les pertes de temps en transport. Un layout mal conçu peut amener à des baisses de rendement significatives.
- **le temps de cycle contrat T_{cc} :** c'est la cadence de production de l'îlot d'assemblage en produits par seconde. Le temps de cycle contrat est fixé dans le cahier des charges.
- **le coût:** le montant est parfois fixé par le client, mais dans tous les cas, le concepteur va chercher à le minimiser. Le coût d'un îlot d'assemblage, en tant que conséquence des choix de conception, peut être interprété comme le coût technologique à payer pour résoudre les différentes contraintes.

Ainsi un îlot d'assemblage doit être capable d'assembler le produit (ou une famille de produits) en respectant le temps de cycle contrat, et la détermination de ses équipements constitutifs est principalement régie par la contrainte économique, le but étant d'obtenir la solution la moins chère qui résolve les autres contraintes.

La caractérisation des équipements se déroule en quatre phases:

- **la détermination des modes opératoires:** les processus d'assemblage sont transformés en processus opératoires qui contiennent non seulement les opérations constitutives, mais aussi les opérations positionnelles (réorientation, positionnement des constituants, alimentation). C'est aussi durant cette phase que sont déterminés les *constituants primaires* (constituants qui restent fixes durant l'opération d'assemblage) et les *constituants secondaires* (constituants mobiles durant l'opération d'assemblage et rapportés sur le constituant primaire). A chaque opération des processus opératoires est alors associé un type de ressources (opérateurs, préhenseurs, distributeurs, équipements spécialisés).
- **l'étude dynamique et le choix des équipements:** sur la base d'un temps d'exécution estimé associé à chaque opération, les processus opératoires sont étudiés du point de vue temporel en fonction:
 - des regroupements possibles des opérations au sein des ressources
 - des regroupements possibles des ressources en postes et cellules

Durant cette phase sont aussi déterminés les types de transferts, les capacités des stocks-tampons, le nombre de palettes, etc. Sur la base des résultats de cette étude dynamique, il est alors possible de décider ce qui peut être automatisé, de choisir les équipements physiques qui constitueront l'îlot et de déterminer leur spécifications techniques.

- **l'implantation spatiale théorique:** une fois les équipements physiques connus, il est possible de déterminer une implantation spatiale des équipements dans l'atelier. Cette implantation doit tenir compte des problèmes ergonomiques (accès aux équipements, incompatibilité entre certains équipements, etc.)
- **la simulation et l'évaluation:** une simulation fine de l'îlot est effectuée qui permet d'évaluer ses performances (temps de cycle, fiabilité)

Le processus de caractérisation des équipements ne s'effectue généralement pas en une seule passe à travers les quatre phases présentées, mais constitue un processus itératif entre les phases qui permet l'affinement des résultats à chaque itération.

Il existe plusieurs outils d'aide à la caractérisation des équipements, qui s'attachent à l'une ou l'autre phase:

- le système AISE [10] du LAB fournit une aide à la détermination des modes opératoires, en offrant la possibilité de manipuler les processus opératoires, et à l'étude dynamique au moyen de réseaux de Petri temporisés.
- le système MARSYAS [53] du LAB fournit une aide au choix des équipements à partir des processus opératoires sur la base de critères économiques et temporels.
- le système développé au CSDL [75]
- le système SIMAS [70] de l'EPFL est un système de simulation qui permet d'évaluer les performances de l'îlot en fonction des taux de pannes, d'arrêt et d'éjection, ainsi que des capacités des stocks-tampons.

Nous citerons encore comme travaux en rapport avec la caractérisation des équipements, les travaux de M. Hongler [44] de l'EPFL sur l'analyse des comportements dynamiques d'une chaîne de production en vue de leur simulation.

5.9 La conception des commandes

La commande des flots d'assemblage se fait à différents niveaux, correspondant aux différents horizons temporels (liste non exhaustive):

- le contrôle temps réel de l'ilot: il s'agit de la commande au niveau des équipements. Il s'appuie, en général, sur des automates programmables.
- le contrôle à court terme: il s'agit de la commande au niveau de la stratégie de pilotage de l'ilot. Elle détermine quel constituant doit passer dans quel poste en fonction de telle ou telle stratégie. Le logiciel SIMAS [70], en plus de permettre la simulation et l'évaluation d'un ilot, peut fournir les commandes de pilotage de l'ilot.
- le contrôle à moyen terme: il s'agit de la commande au niveau de la constitution des lots de fabrication, en fonction des commandes passées par les clients. Il s'agit de l'ordonnancement des lots de fabrication.

La commande de système de production est le premier domaine de la productive dans lequel des solutions multi-agents sont proposées:

- F. R. Gouveia et E. da Costa Oliveira [38] proposent une architecture multi-agents pour la gestion complète d'un atelier. Elle est basée sur trois types d'agents: les agents ouvriers, les agents superviseurs et les agents espions. Les agents ouvriers sont les machines de l'atelier et ils sont responsables du contrôle à court terme. Les agents superviseurs sont responsables du contrôle à moyen terme (planification, ordonnancement) et commandent aux agents ouvriers. Les agents espions sont chargés de l'observation des performances du système et de fournir les informations nécessaires aux travaux des autres agents.
- P. J. Gmytrasiewicz et al. [34] proposent une modélisation orientée agent pour les machines de production. Cette modélisation transforme une machine en agent capable de prendre des décisions, basées sur la situation locale courante dans laquelle elle se trouve (états des pièces en entrée et accointances du moment), afin d'exécuter l'action la plus profitable pour le processus de production en cours.
- T. Daouas [17, 16] propose un système multi-agents d'ordonnancement, basé sur le modèle de l'éco-résolution, où les commandes client et les ressources interagissent de manière à minimiser les délais et à satisfaire les contraintes de précedence.

5.10 Conclusion

La conception de systèmes d'assemblage s'articule principalement autour de la détermination des processus d'assemblage et de la caractérisation des équipements.

Un processus d'assemblage est une description fonctionnelle d'une manière d'assembler un produit. Il est formé d'un ensemble ordonné de tâches d'assemblage, amenant du pré-produit au produit fini. Un processus d'assemblage est admissible lorsque toutes les tâches qui le composent répondent aux contraintes-produit (géométrie des constituants, accessibilité, stabilité). La détermination des processus d'assemblage admissibles d'un produit se fait, en général, par décomposition d'un modèle du produit (résultat d'une phase d'analyse du problème) et contrôle des contraintes. On sélectionne ensuite parmi ces processus, les processus d'assemblage répondant au mieux à des critères stratégiques de production.

Par analogie avec la conception en général, la détermination et la sélection des processus d'assemblage constituent le traitement de l'espace *fonctionnel* de la conception des systèmes d'assemblage.

La caractérisation des équipements consiste dans la détermination des équipements constitutifs d'un flot d'assemblage et de leur organisation en son sein (postes, cellules, flot). Elle est basée sur un processus opératoire (processus d'assemblage complété de tâches logistiques), regroupant les modes opératoires retenus pour la réalisation des tâches, et doit répondre aux contraintes-ressources (temps de cycle contrat, coût, flexibilité, fiabilité, ergonomie).

Par analogie avec la conception en général, la caractérisation des équipements constitue le traitement de l'espace *physique* de la conception des systèmes d'assemblage.

La spécification de l'flot d'assemblage conçu est alors complétée par ses différents niveaux de commandes, qui semble être le seul domaine de la conception des systèmes d'assemblage où apparaissent des solutions multi-agents.

Partie II

Le système AMACOIA

Chapitre 6

Présentation générale

6.1 Introduction

Le système AMACOIA¹ est un outil d'aide pour la conception de systèmes d'assemblage. Il offre une aide à l'évaluation et à la sélection des processus d'assemblage admissibles, ainsi qu'à la caractérisation des équipements.

La conception des systèmes d'assemblage y est abordée comme un problème combinatoire, ayant pour objet deux espaces de recherche distincts, mais liés:

- la détermination et la sélection des processus d'assemblage consistent à construire, puis à parcourir l'*espace fonctionnel* de la conception des systèmes d'assemblage, que forme l'ensemble des processus d'assemblage du produit.
- la caractérisation des équipements consiste à parcourir l'*espace physique* de la conception des systèmes d'assemblage, que forme l'ensemble des îlots d'assemblage.

Au chapitre 4, nous avons vu qu'il n'existe pas de relation déductive directe entre fonctions à réaliser et la structure physique. Il s'avère donc impossible de traiter automatiquement les espaces fonctionnel et physique de la conception. Par contre il s'avère possible d'appréhender ces espaces au travers de modèles (manipulables informatiquement) avec pour effet de restreindre les espaces de recherche. Sur la base de ces modèles (comportements recherchés et comportement productibles), il est alors possible de structurer et d'évaluer les étapes du processus de conception.

Ainsi, le système AMACOIA n'est pas un système de conception (puisque c'est impossible), mais un outil d'aide à la conception, offrant les modèles sur lesquels peut s'appuyer le concepteur pour structurer et évaluer ses choix.

Dans le chapitre 5, nous avons vu que l'assemblage d'un produit représente jusqu'à 70 pour cent de son coût de fabrication. Dans le but de réduire l'impact du coût de l'assemblage sur le coût du produit, le système AMACOIA a été développé pour pouvoir évaluer et optimiser le coût d'assemblage d'un produit. Ce coût est représenté par le coût d'un îlot d'assemblage capable d'assembler le produit.

¹AMACOIA: Approche Multi-Agents de la COncption d'Installations d'Assemblage

Ainsi, le système AMACOIA a été développé comme un outil à l'usage principal des concepteurs de systèmes d'assemblage pour qu'ils puissent évaluer les conséquences économiques de leurs choix de conception. En tant que tel, le système est alors utilisé, soit comme support à la créativité de l'ingénieur, afin d'explorer les différentes solutions intéressantes, soit pour obtenir rapidement une première spécification d'un îlot d'assemblage.

D'autre part, 75 pour cent du coût d'un produit sont définis au moment de sa conception. En utilisant AMACOIA, les concepteurs de produit peuvent évaluer les implications économiques de leur choix de conception sur le coût d'assemblage du produit. En procédant ainsi, ils peuvent réduire encore le coût du produit, en tenant compte, dans la phase de conception, des éléments qui constitueront la plus grande part de son coût final.

Afin de répondre aux besoins des concepteurs de systèmes d'assemblage, comme à ceux des concepteurs de produits, AMACOIA a été conçu pour s'adapter aux exigences et à l'expertise de l'utilisateur. Ainsi, le principe de base du système est la répartition des tâches entre l'expert humain et l'ordinateur.

Ce principe du partage des tâches est basé sur le fait suivant:

- les capacités d'analyse d'un expert humain sont supérieures (en qualité et en temps) à celles de l'ordinateur, alors que les capacités de calcul de l'ordinateur sont supérieures à celles de l'expert humain.

Dès lors les capacités d'analyse de l'expert humain sont utilisées pour restreindre, contraindre et définir l'espace de recherche sur lequel il souhaite travailler, alors que l'ordinateur est utilisé pour manipuler et explorer cet espace, sous la direction de l'utilisateur.

L'expert humain peut donc configurer le système, selon les capacités de raisonnement d'AMACOIA, la difficulté et les contraintes du problème à résoudre et le degré de précision souhaité pour les résultats. L'utilisateur fixant ainsi la répartition des tâches entre lui et l'ordinateur, le système s'adapte alors aux conditions imposées par l'humain.

Le but de ce chapitre est de présenter de manière générale le système AMACOIA et les méthodes utilisées, les détails du système étant présentés dans les chapitres suivants.

6.2 Le processus de conception

Le chapitre 4 a montré que la conception peut être modélisée comme un parcours à travers deux espaces de recherche: l'espace fonctionnel, résultant de l'expression des fonctions en comportements recherchés, et l'espace physique, résultant de la synthèse des comportements productibles en une structure physique. La conception de systèmes d'assemblage, présentée au chapitre 5, constitue une instantiation de ces deux espaces et de leur parcours (cf. figure 6.1).

L'espace fonctionnel de la conception des systèmes d'assemblage est constitué de l'ensemble des processus d'assemblage du produit considéré. L'espace des solutions possibles de cet espace fonctionnel est constitué de l'ensemble des processus d'assemblage admissibles qui résolvent les contraintes-produit (géométrie, accessibilité et stabilité). L'optimum est le processus d'assemblage admissible qui répond au mieux aux critères stratégiques de production tels que décrits au point 5.7.2.

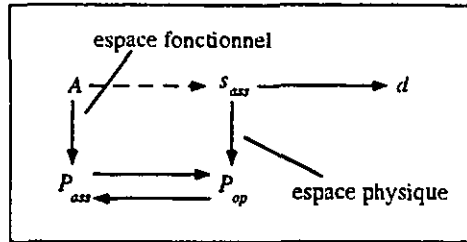


Figure 6.1: Les éléments de la conception des systèmes d'assemblage. A , la fonction "assembler un produit"; P_{ass} , l'ensemble des processus d'assemblage; P_{op} , l'ensemble des processus opératoires réalisés par le système; s_{ass} , le système d'assemblage; d , la description.

L'espace physique de la conception des systèmes d'assemblage est constitué de l'ensemble des flots d'assemblage qui réalisent les processus d'assemblage du produit. L'espace des solutions possibles de cet espace physique est constitué de l'ensemble des flots d'assemblage qui résolvent les contraintes-ressources (temps de cycle contrat, flexibilité, fiabilité). L'optimum est l'ilot d'assemblage le moins cher.

L'espace physique est construit sur la base de l'espace fonctionnel: un point de l'espace fonctionnel (un processus d'assemblage) détermine alors un sous-espace de l'espace physique (dans le sens que plusieurs ilots d'assemblage peuvent être conçus sur la base du même processus d'assemblage). L'espace physique n'est pas isomorphe à l'espace fonctionnel.

Comme le processus de conception décrit au point 5.5, le processus de conception implanté par AMACIOA distingue le traitement de l'espace fonctionnel (caractérisation des processus d'assemblage) du traitement de l'espace physique (caractérisation des équipements).

Le traitement de l'espace fonctionnel est basé sur une nouvelle modélisation du produit, appelée *l'arbre des références opératoires*². Ce modèle est une modélisation opératoire qui englobe non seulement les caractéristiques d'assemblage du produit (composants et caractéristiques géométriques, physiques et complémentaires), mais aussi la représentation de certains choix technologiques sur la réalisation des caractéristiques: choix des composants primaires et secondaires, directions de travail, etc.

L'utilisation du modèle de l'arbre des références opératoires a pour conséquence de réduire l'espace de recherche, tout en garantissant de rester dans l'espace des solutions. Il permet également à l'utilisateur d'y introduire sa propre expérience et sa propre vision du problème.

Le parcours de l'espace fonctionnel et le parcours de l'espace physique utilisent des systèmes multi-agents en combinaison avec la méthode du recuit simulé. Cette combinaison offre les avantages suivants:

- les systèmes multi-agents, basés sur le modèle de l'éco-résolution, permettent de gérer la complexité des interdépendances entre critères géométriques, techniques, économiques et temporels lors du parcours des espaces, les conflits se réglant par négociation et coopération entre ces agents.

²L'arbre des références opératoires est décrit en détail au point 7.1.

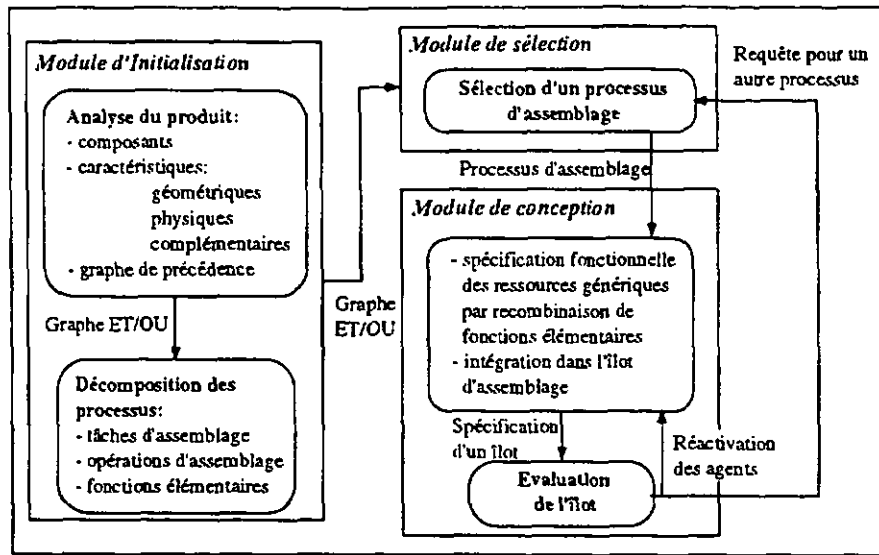


Figure 6.2: La méthode de conception utilisée dans AMACIOA et les trois modules qui l'implémentent.

- le recuit simulé permet la recherche d'une solution proche de l'optimum en évitant le piège des optima locaux.

L'approche multi-agents offre également beaucoup de souplesse face aux modifications dynamiques des données au cours du processus de conception (ajout ou suppression de contraintes par l'utilisateur), le système recherchant immédiatement un nouvel état d'équilibre à partir de la solution courante.

6.2.1 L'espace fonctionnel

L'espace fonctionnel traité dans AMACIOA est un sous-espace de l'espace complet que constituent tous les processus d'assemblage possibles d'un produit: il n'est constitué que de processus d'assemblage admissibles qui résolvent les contraintes-produit. Ainsi, tout point de l'espace fonctionnel traité est solution. De plus, il n'est pas constitué de tous les processus d'assemblage admissibles du produit, mais de ceux qui répondent déjà à certains critères stratégiques, laissés à la préférence de l'utilisateur.

Ces restrictions viennent du fait que l'espace fonctionnel traité n'est pas construit par le système, mais en constitue une entrée: il est donné par l'utilisateur sous la forme de l'arbre des références opératoires et d'un graphe de précédence. Par hypothèse, nous considérons que les contraintes représentées dans le graphe de précédence constituent aussi bien des solutions aux contraintes-produit (géométrie, accessibilité et stabilité) que l'expression des critères stratégiques de production retenus par l'utilisateur.

De fait, l'optimum de l'espace fonctionnel recherché dans AMACIOA n'est plus le processus d'assemblage répondant au mieux aux critères stratégiques, mais le processus d'assemblage dont les tâches sont les plus simples à exécuter.

Le traitement de l'espace fonctionnel se déroule en deux phases:

1. l'expression de l'espace fonctionnel "complet" sous une forme qui permette l'évaluation des processus d'assemblage. L'espace fonctionnel, exprimé dans le graphe de précédence et l'arbre des références opératoires, est transformé en un graphe ET/OU, où apparaissent les tâches et les constituants impliqués dans chaque processus. Cette transformation se fait par décomposition de l'arbre des références opératoires en contrôlant les contraintes de précédence. Chaque processus est alors découpé fonctionnellement, des tâches d'assemblage en fonctions élémentaires.

Cette étape est effectuée par le *module d'initialisation* du système AMACOIA décrit au chapitre 7).

2. l'exploration de l'espace fonctionnel obtenu, à la recherche de son optimum. Cette recherche est implantée dans un système multi-agents (SMA) et contrôlée par un recuit simulé.

Cette étape est effectuée par le *module de sélection* du système AMACOIA décrit au chapitre 8).

6.2.2 L'espace physique

L'espace physique des flots d'assemblage est constitué de plusieurs sous-espaces définis par chacun des processus d'assemblage de l'espace fonctionnel: plusieurs flots d'assemblage peuvent réaliser le même processus d'assemblage. Les sous-espaces ainsi définis ne sont pas forcément indépendants mais peuvent se chevaucher, un flot d'assemblage pouvant réaliser plusieurs processus d'assemblage.

A l'inverse de l'espace fonctionnel, l'espace physique n'est pas donné a priori, mais est construit dynamiquement au fur et à mesure de son parcours. Un point de l'espace physique, que représente un flot d'assemblage, est obtenu par recombinaison des fonctions élémentaires d'un processus d'assemblage en équipements, postes, cellules et flot en satisfaisant les contraintes-ressources.

Le parcours de l'espace physique se fait sous-espace par sous-espace en recherchant l'optimum, que constitue l'flot d'assemblage le moins cher, de chacun de ces sous-espaces.

Le traitement de l'espace physique est implanté dans un système multi-agents. Chaque agent est spécialisé dans la résolution d'une partie du problème et la spécification de l'flot d'assemblage est obtenue par négociation et auto-organisation de ceux-ci. De cette manière, le SMA génère les points de l'espace physique. L'exploration de l'espace est contrôlée par le recuit simulé évaluant le coût de l'flot généré.

Ce système multi-agents constitue le *module de conception* du système AMACOIA décrit au chapitre 9.

6.2.3 L'utilisateur

Comme AMACOIA n'est pas un système de conception, mais un outil d'aide à la conception, offrant les modèles sur lesquels peut s'appuyer le concepteur pour structurer et évaluer ses choix, le rôle de l'utilisateur est donc très important.

L'utilisateur est à la base du processus de conception, puisqu'il décrit au système l'espace fonctionnel qu'il souhaite explorer, mais ensuite, en fonction de son expertise, de la précision des résultats souhaitée, des compétences de

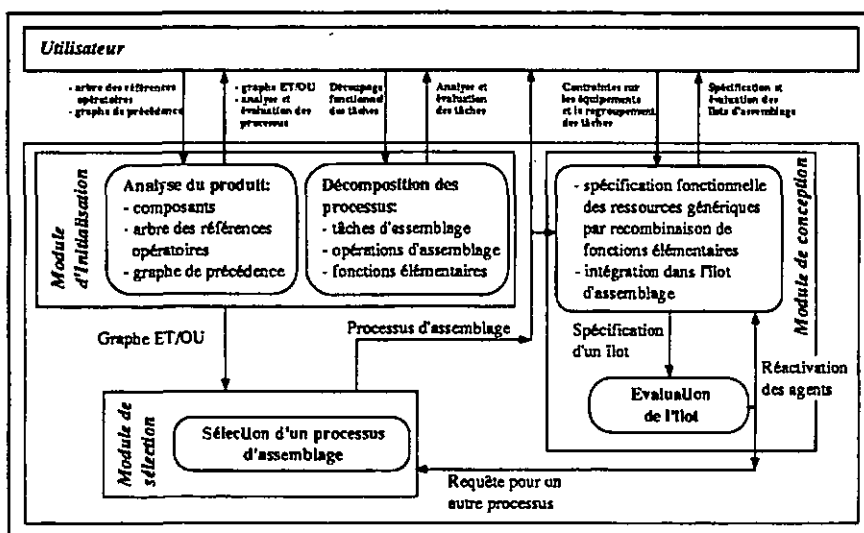


Figure 6.3: Interactions entre l'utilisateur et le système AMACOIA

base d'AMACOIA ou des résultats fournis par le système, il peut modifier ou influencer le processus de conception. L'utilisateur fixe ainsi les données et les contraintes, et le système s'adapte alors aux conditions imposées par l'humain.

L'utilisateur peut interagir avec le système de trois façons (cf. figure 6.3):

1. l'utilisateur fixe l'espace fonctionnel qu'il veut explorer, au moment où il décrit au système l'arbre des références opératoires et le graphe de précedence. Ces descriptions sont les résultats de son analyse du problème et constituent le point de départ de tout le processus de conception. AMACOIA transforme alors ces descriptions en graphe ET/OU que l'utilisateur peut consulter.
2. au cours du découpage fonctionnel des tâches des processus d'assemblage, l'utilisateur a la possibilité de décrire son propre découpage en opérations et fonctions élémentaires. En procédant ainsi, il peut introduire sa propre expérience et/ou pallier aux carences du système. Les résultats du découpage fonctionnel (fait par l'utilisateur ou automatiquement par AMACOIA), sont évalués et l'utilisateur peut les consulter.
3. au moment de la conception des îlots, l'utilisateur peut imposer des solutions, en termes d'équipements et de regroupements de tâches. Là aussi, il introduit sa propre expérience et peut ainsi améliorer les performances du système. Les îlots d'assemblage déterminés par AMACOIA, en tenant compte des éventuelles contraintes de l'utilisateur, peuvent être analysés et consultés par l'utilisateur.

Par ces interactions, l'utilisateur peut délimiter les sous-problèmes qu'il souhaite lui-même résoudre et les sous-problèmes que doit résoudre AMACOIA, dont les agents sont dotés de capacités (limitées) de raisonnement.

Les systèmes multi-agents favorisent cette approche, puisque, par nature, ils découpent déjà le problème en sous-problèmes et qu'ils offrent les mécanismes

d'interactions entre ces sous-problèmes.

Le point de vue de l'utilisateur est présenté plus en détails au chapitre 10, en même temps que l'exemple.

6.3 Génération des espaces

La sélection d'un processus d'assemblage (constituant un point de l'espace fonctionnel) et la génération d'un flot d'assemblage (constituant un point de l'espace physique) sont obtenues par la dynamique résultant de l'interaction des agents.

Les deux systèmes multi-agents qui constituent le module de sélection et le module de conception d'AMACOIA, sont basés sur le modèle de l'éco-résolution, tel que décrit au chapitre 2.

Selon ce modèle, les populations d'agents des SMA sont constituées d'agents réactifs dont le comportement est guidé par la recherche de satisfaction: la réception d'un message en provenance d'une accointance a pour effet de modifier l'état de satisfaction de l'agent, et il déclenche alors une réaction appropriée à la modification, afin de se rapprocher de ses conditions de satisfaction. La communication se fait par envoi de messages, soit en mode point à point, soit par diffusion.

Chaque agent d'AMACOIA a pour objectifs, donnés par ses conditions de satisfaction, d'élaborer un élément de la structure qui lui est propre, et d'intégrer cet élément dans la structure (par exemple: dans le module de conception, un agent-poste est chargé d'élaborer un des postes d'assemblage et de l'intégrer à l'ilot dans le cadre d'une cellule). L'élaboration de l'élément de structure de l'agent se fait par l'intégration et la combinaison d'éléments fournis par les autres agents (un agent-cellule, pour élaborer une cellule d'assemblage, combine les postes fournis par les agents-postes).

Le comportement d'un agent d'AMACOIA peut être résumé par le comportement réflexe suivant:

$R_a = \langle C_{sat}, R, I \rangle$ avec:

- C_{sat} , les conditions de satisfaction de l'agent qui sont:
 - *Condition de satisfaction technologique*: la partie de structure créée par l'agent doit former une combinaison technologiquement cohérente d'éléments (la définition de la cohérence technologique est fonction de la structure créée par l'agent).
 - *Condition de satisfaction d'intégration*: la structure créée par l'agent doit être intégrée, en tant qu'élément, dans une structure plus grande.
- R , l'ensemble des réactions de l'agent. Chaque réaction est déclenchée par un message en provenance d'une accointance. Une réaction se déroule en général de la manière suivante:
 1. *Modification de la structure*: selon le message reçu, l'agent cherche à modifier sa structure de manière à se rapprocher de sa condition technologique.
 2. *Contrôle de la condition d'intégration*: si cette condition de satisfaction n'est pas remplie, l'agent envoie une requête de placement à une accointance appropriée.

- *I*, l'information diffusée par un agent à ses accointances. Elle est constituée des décisions prises par l'agent durant sa réaction et des requêtes de placement.

Ainsi, la condition de satisfaction d'intégration pousse l'agent à envoyer une requête de placement à l'une de ses accointances susceptible de l'intégrer, dont la condition de satisfaction technologique, lui permettra alors de décider si il accepte ou non cette requête.

Les échanges de requêtes, acceptations et rejets s'arrêtent lorsque tous les agents sont satisfaits. Le point de l'espace de recherche est ainsi généré.

Dans le module de sélection, les agents sont générés par le système. Par contre, dans le module de conception, le modèle de l'éco-résolution a été étendu en introduisant la création et la destruction dynamiques d'agents: certains agents sont générés par les autres agents au fur et à mesure de leur besoin. Ces agents créés dynamiquement cherchent, en plus de leur satisfaction, à maintenir une condition d'existence qui, si elle ne plus être remplie à un moment donné, les pousse au suicide.

6.4 Parcours des espaces

L'objectif du parcours de l'espace fonctionnel et du parcours de l'espace physique est d'obtenir, dans le premier cas, le processus d'assemblage dont les tâches sont les plus simples à réaliser, et dans le second cas, l'ilot d'assemblage le moins cher.

Ces problèmes d'optimisation sont traités par la méthode du recuit simulé, décrite ci-dessous, et intégrée dans la dynamique des systèmes multi-agents.

6.4.1 Le recuit simulé

Le recuit simulé est un algorithme stochastique d'optimisation combinatoire développé par S. Kirkpatrick [45].

L'objectif, en optimisation combinatoire est de choisir parmi l'ensemble des configurations possibles pour le problème qu'on a à traiter, une configuration qui satisfasse les contraintes et optimise les critères. Ces critères sont généralement représentés par une fonction appelée *fonction coût*.

Le recuit simulé est basé sur la simulation du processus du recuit thermique. Le recuit thermique est une méthode métallurgique d'élimination des défauts dans un cristal.

En effet, au cours de sa solidification, la structure cristalline du métal obtenue correspond à l'état d'énergie minimale que peuvent prendre ses molécules constitutives. L'état d'énergie du cristal obtenu (donc sa structure) dépend du profil de refroidissement: si l'abaissement de la température est trop rapide, les molécules n'ont pas le temps de former un cristal parfait (correspondant au minimum global d'énergie) et il persiste des défauts. Le recuit thermique consiste à réchauffer le cristal (on élève son énergie) pour lui faire franchir la barrière énergétique qui sépare l'optimum local où il se trouve (cristal avec défauts) de l'optimum global (cristal parfait). Puis on reprend le refroidissement de manière lente, en marquant des paliers de température de durée suffisante, afin de lui permettre d'atteindre, quitte à recuire plusieurs fois, la structure optimale (cf figure 6.4).

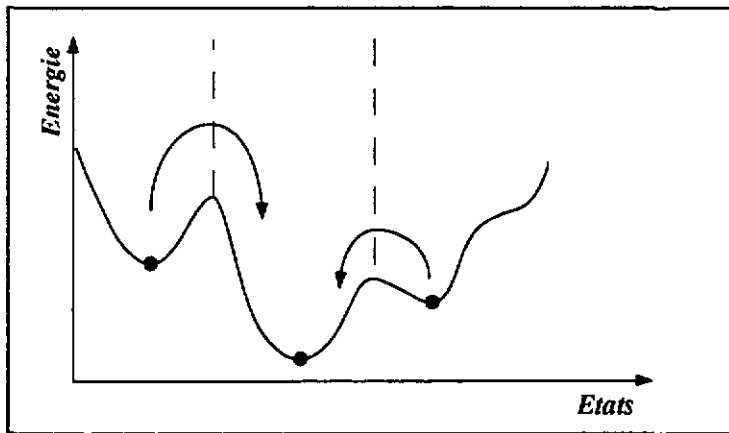


Figure 6.4: A température non nulle et d'autant plus fréquemment qu'elle est élevée, le caractère aléatoire des changements de configuration peut permettre au système de remonter la pente et ainsi d'augmenter son énergie, l'idée étant que ces mouvements localement désavantageux permettront de sauter les barrières séparant les vallées et ainsi échapper aux minima locaux.

En identifiant fonction coût et énergie, et configurations et états physiques, on peut proposer une méthode de résolution des problèmes d'optimisation combinatoire basée sur la simulation du processus du recuit thermique: c'est le recuit simulé. La température est alors un paramètre de contrôle qui dépend du problème à traiter.

Le processus de déplacements aléatoires dans l'espace des configurations est quant à lui caractérisé par la probabilité de transition d'un état à l'autre.

La probabilité de transition de l'état i à l'état j , à l'instant k , est définie comme suit:

$$\forall i, j \in E: P_{ij}(T_k) = \begin{cases} G_{ij}(T_k) * A_{ij}(T_k) & \text{si } i \neq j \\ 1 - \sum_{m \in E, m \neq i} P_{im}(T_k) & \text{si } i = j \end{cases}$$

avec:

- G_{ij} , la probabilité de génération de j à partir de i ,
- A_{ij} , la probabilité d'acceptation de j (une fois généré à partir de i),
- T_k , la température à l'instant k ,
- E , l'espace des états.

La probabilité de génération est de la forme:

$$\forall i, j \in E: G_{ij}(T_k) = \frac{1}{|V_i|} \chi_i(j)$$

avec:

- V_i , le voisinage de l'état i , c'est-à-dire l'ensemble des états atteignables depuis i ,

- $\chi_i(j)$, la fonction d'adjacence de i .

Si e désigne l'énergie (ou la fonction coût) à minimiser, la probabilité d'acceptation s'écrit sous la forme:

$$\forall i, j \in E : A_{ij}(T_k) = \begin{cases} 1 & \text{si } e(j) \leq e(i) \\ \exp\left(\frac{e(i)-e(j)}{T_k}\right) & \text{si } e(j) > e(i) \end{cases}$$

En pratique, on implante le recuit simulé selon un processus itératif, dont une étape consiste à générer un nouvel état à partir de l'état courant, puis, selon sa probabilité d'acceptation, à accepter ou non ce nouvel état comme le nouvel état courant.

6.4.2 Dynamique

L'utilisation pratique du recuit simulé requiert:

- une structure de données représentant l'espace des états; dans AMACOIA, cette structure est constituée des systèmes multi-agents.
- une structure de voisinage représentant pour chaque état l'ensemble des états possibles qui peuvent être atteints en une étape; dans AMACOIA, elle est donnée par la dynamique des systèmes multi-agents, telle que décrite au point 6.3.
- une fonction coût à optimiser; dans AMACOIA, il y a deux fonctions coût: une pour l'espace fonctionnel, la difficulté d'exécution d'un processus, et une pour l'espace physique, le coût d'un flot d'assemblage.
- une température et un schéma de refroidissement; dans AMACOIA, nous avons choisi un profil de refroidissement à décroissance arithmétique caractérisé par les trois paramètres: T_{init} , la température initiale, pa_{init} , la longueur du palier durant lequel T est constante et a , la hauteur de la décroissance de T (cf. figure 6.5).

Remarque: ce profil de refroidissement semble être le plus utilisé en recuit simulé, c'est pourquoi nous l'avons choisi. Nous n'avons pas testé d'autres profils, ni les influences qu'ils peuvent avoir sur les performances du système.

Dans AMACOIA, le parcours de l'espace de recherche est géré par un seul agent (l'agent-processus dans le module de sélection et l'agent-flot dans le module de conception). Une fois qu'un point de l'espace, construit selon la dynamique décrite au point 6.3, est obtenu, cet agent applique le recuit simulé. Sur la base de cette évaluation, l'agent accepte alors ou non le point généré comme le nouveau point courant. Puis, sur la base du point courant (qu'il ait changé ou non), il réactive aléatoirement les autres agents du système, avec pour conséquence la perte de leur satisfaction: le système va tendre alors vers une nouvelle solution. La trajectoire du parcours de l'espace de recherche est ainsi constituée des points courants successifs.

Le parcours s'arrête lorsque, au moment de l'application du recuit simulé, la tolérance au changement T^{α} de l'agent responsable du recuit est nulle.

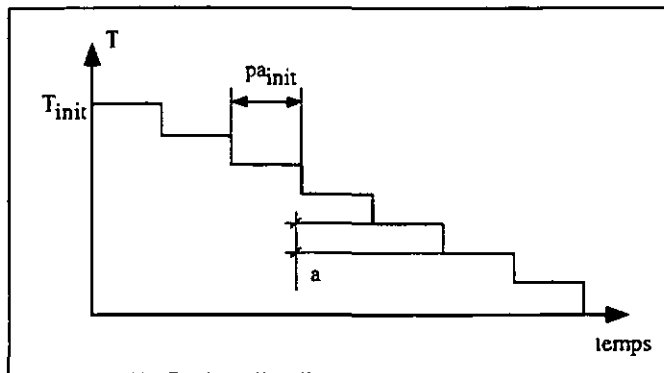


Figure 6.5: Profil de refroidissement de la température T utilisé dans AMACOIA: T_{init} , température initiale; pa_{init} , longueur du palier; a , hauteur de la décroissance.

6.5 Conclusion

Le système AMACOIA est un outil d'aide à la conception des systèmes d'assemblage. La conception des systèmes d'assemblage y est abordée comme un problème d'optimisation combinatoire ayant pour objet deux espaces de recherche distincts, mais liés:

- l'espace fonctionnel, constitué des processus d'assemblage du produit. L'optimum recherché est le processus d'assemblage dont les tâches sont les plus simples à réaliser (d'un point de vue fonctionnel).
- l'espace physique, constitué des flots d'assemblage. L'optimum recherché est l'flot d'assemblage le moins cher.

Comme il s'avère impossible de traiter automatiquement ces espaces, puisqu'il n'y a pas de relation directe entre fonctions à réaliser (assembler un produit) et structure physique (l'flot d'assemblage), le système met à disposition de l'utilisateur les modèles (arbre des références opératoires, systèmes multi-agents) sur lesquels il peut s'appuyer pour structurer et évaluer les étapes du processus de conception.

Ainsi, le rôle de l'utilisateur est très important: il restreint, contraint et définit les espaces de recherche sur lequel il souhaite travailler, alors que le système est utilisé pour manipuler et explorer ces espaces, sous la direction de l'utilisateur.

Le système AMACOIA est constitué de trois modules:

- un module d'initialisation dont le rôle est de transformer l'espace fonctionnel, décrit par l'utilisateur par l'arbre des références opératoires et un graphe de précedence, sous une forme utilisable par les autres modules.
- un module de sélection, formé d'un système multi-agents chargé d'explorer l'espace fonctionnel.
- un module de conception, formé d'un système multi-agents chargé d'explorer l'espace physique.

Le problème d'optimisation combinatoire est résolu au moyen de la méthode du recuit simulé. Le recuit simulé est implanté par les systèmes multi-agents, basés sur le modèle de l'éco-résolution. La dynamique de ces agents permet de générer les points des espaces parcourus.

Chapitre 7

Le module d'initialisation

Le module d'initialisation a pour objet l'espace fonctionnel de la conception des systèmes d'assemblage. Ses objectifs sont:

1. la saisie du sous-espace de l'espace fonctionnel que souhaite explorer l'utilisateur.
2. la transformation de ce sous-espace en une forme utilisable par les autres modules d'AMACOIA.

AMACOIA ne travaille pas sur l'espace fonctionnel complet de la conception de systèmes d'assemblage, constitué de tous les processus d'assemblages possibles pour assembler le produit. Son espace de recherche fonctionnel est choisi par l'utilisateur et il est constitué de processus d'assemblage admissibles qui satisfont aux contraintes-produit.

L'utilisateur décrit cet espace de recherche au moyen d'un modèle du produit, l'arbre des références opératoires (section 7.1), et d'un graphe de précédence (section 7.2), qui forment les entrées du système.

La description de l'utilisateur est ensuite transformée par le système en un graphe ET/OU (section 7.3) qui sera utilisé par le module de sélection d'AMACOIA pour évaluer les processus d'assemblage retenus.

Le module d'initialisation précise ensuite la description des tâches composant le graphe ET/OU par la décomposition fonctionnelle (section 7.4) de chacune d'elles. Les résultats de cette décomposition seront utilisés dans le module de conception d'AMACOIA.

7.1 L'arbre des références opératoires

Le modèle du produit utilisé dans AMACOIA est un modèle opératoire¹. Il regroupe donc toutes les caractéristiques (géométriques, physiques, complémentaires) du produit qui devront être réalisées au cours du processus d'assemblage, mais aussi certains choix technologiques pour leur réalisation: le choix des références géométriques (pièces) par rapport auxquelles seront réalisées les caractéristiques, et la direction principale des trajectoires de travail.

¹cf. 5.6.1

A chaque caractéristique est associé un composant élémentaire du produit. Ce composant, appelé le *récepteur*, est la référence géométrique par rapport à laquelle est décrite et réalisée la caractéristique. Une caractéristique n'a qu'un et un seul récepteur.

A une caractéristique géométrique est associé un second composant élémentaire du produit, appelé simplement le *composant*. Le composant est le composant élémentaire qui est assemblé au récepteur au moment de la réalisation de la caractéristique géométrique. Pour une caractéristique géométrique, la notion de récepteur et composant peut être rapprochée de la notion de constituant primaire (récepteur) et constituant secondaire (composant) telle que décrite au point 5.8.

A chaque caractéristique est également associée une direction principale de travail, décrite par rapport au récepteur. La direction de travail correspond à la direction générale des trajectoires d'insertion, de mise en contact, ou d'approche des outils.

Un produit p^j est décrit par le couple:

$$p^j = \langle C_e^j, C^j \rangle \text{ avec:}$$

- C_e^j , l'ensemble des composants élémentaires (pièces) du produit. Un composant élémentaire c_e^i est décrit par le triplet:

$$c_e^i = \langle D^i, F^i, G^i \rangle \text{ avec:}$$

- D^i , l'ensemble des cotes du composant avec leurs tolérances.
- F^i , le sous-ensemble des faces élémentaires du composant constitué des faces importantes pour l'assemblage: faces de contact, faces de saisie possibles, faces de positionnement possibles. Elles sont décrites sur la base des éléments de D^i .
- G^i , l'ensemble des relations géométriques entre les faces de F^i . Elles décrivent les positions et orientations des faces par rapport au repère de la pièce. Elles sont également définies sur la base des éléments de D^i .

- C^j , l'ensemble des caractéristiques du produit.

$$C^j = \{G_a^j, \Phi^j, C_c^j\} \text{ avec:}$$

- G_a^j , l'ensemble des caractéristiques géométriques d'assemblage du produit. Une caractéristique géométrique d'assemblage est une relation géométrique décrivant la position et l'orientation entre le composant et le récepteur. Une caractéristique géométrique d'assemblage ne correspond pas forcément à un contact entre les deux pièces.

Une caractéristique géométrique d'assemblage est décrite par le quintuplet:

$$g_a = \langle f_r, f_c, M, ddl, dir \rangle \text{ avec:}$$

f_r , la face du récepteur selon laquelle est décrite g ; f_c , la face du composant; M , une matrice de coordonnées homogènes décrivant la position et l'orientation du repère de f_c dans celui de f_r ; ddl , une matrice 3×4 des degrés de liberté décrivant les degrés de liberté restants au composant, par rapport au récepteur après création de g_a ; dir , la direction générale d'assemblage décrite par rapport au repère de f_r .

- Φ^j , l'ensemble des caractéristiques physiques du produit. Ce sont les procédés de solidarisation utilisés pour bloquer les éléments de G_a^j .

Une caractéristique physique est décrite par le quadruplet:

$\phi = \langle f_r, G, proc, dir \rangle$ avec:

f_r , la face du récepteur par rapport à laquelle sera réalisée ϕ ; G , l'ensemble des caractéristiques géométriques d'assemblage bloquées par ϕ ; *proc*, le procédé de solidarisation utilisé: sans composant d'attachement (chassage, clippage), avec composant discret (vis, rivet), avec composant conditionné (colle, soudage); *dir*, la direction, décrite dans le repère de f_r , selon laquelle sera réalisée ϕ .

- C_c^j , l'ensemble des caractéristiques complémentaires du produit.

Une caractéristique complémentaire est décrite par le triplet:

$c_c = \langle f_r, sub, dir \rangle$ avec:

f_r , la face du récepteur par rapport à laquelle sera réalisée c_c ; *sub*, le sous-assemblage minimal qui doit être assemblé pour pouvoir réaliser c_c ; *dir*, la direction, décrite dans le repère de f_r , selon laquelle sera réalisée c_c .

Ce modèle peut être représenté sous la forme d'un graphe $A_{r_0} = \langle S, A \rangle$ tel que:

- S , l'ensemble de ses sommets, correspond à l'ensemble des composants élémentaires et des caractéristiques $\in C_c^j \cup C^j$;
- A , l'ensemble de ses arcs, correspond à l'ensemble des relations entre caractéristiques et composants élémentaires indiquant soit "a pour récepteur", soit "est le composant de". Ainsi les arcs sont orientés soit de la caractéristique vers son récepteur, soit du composant vers sa caractéristique géométrique d'assemblage.

A_{r_0} est un arbre, dû à l'hypothèse du récepteur unique, et il est appelé *l'arbre des références opératoires*.

7.1.1 Exemple

Selon la modélisation présentée, la valve de vélo est décrite par le couple $\langle C_e^{valve}, C^{valve} \rangle$. L'ensemble des composants élémentaires vaut: $C_e^{valve} = \{corps, guide, soupape, joint, écrou\}$. L'ensemble des caractéristiques C^{valve} est constitué de:

- G_a^{valve} , l'ensemble de ses caractéristiques géométriques d'assemblage:
 $G_a^{valve} = \{g_a^1, g_a^2, g_a^3, g_a^4\}$ avec g_a^1 la caractéristique géométrique d'assemblage entre le composant *joint* et le récepteur *soupape*, g_a^2 , celle entre la *soupape* et le *corps*, g_a^3 , celle entre le *guide* et le *corps* et g_a^4 , celle entre l'*écrou* et la *soupape*.
- Φ^{valve} , l'ensemble de ses caractéristiques physiques: $\Phi^{valve} = \{\phi^1, \phi^2, \phi^3\}$ avec ϕ^1 le chassage du *joint* sur la *soupape*, ϕ^2 , le chassage du *guide* dans le *corps* et ϕ^3 , le vissage de l'*écrou* sur la *soupape*.

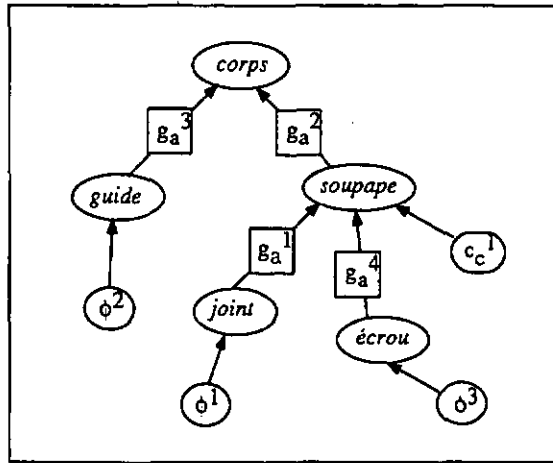


Figure 7.1: Arbre des références opératoires de la valve de vélo

- C_c^{valve} , l'ensemble des caractéristiques complémentaires: $C_c^{valve} = \{c_c^1\}$ avec c_c^1 , le sertissage de la soupape.

L'arbre des références opératoires de la valve est donné à la figure 7.1.

7.2 Le graphe de précedence

Le graphe de précedence, noté G_p , regroupe les contraintes de précedence pour la réalisation des caractéristiques. Le graphe de précedence est utilisé par l'expert, en complément de l'arbre des références opératoires, pour décrire l'espace fonctionnel qu'il a choisi d'explorer. Certaines des contraintes peuvent être déduites de A_{ro} (par exemple, une caractéristique physique doit être réalisée après la (les) caractéristique(s) géométrique(s) d'assemblage qu'elle fixe), mais les contraintes de précedence représentant les solutions aux contraintes-produit ou découlant des critères stratégiques de production retenus par l'utilisateur ne peuvent être déduites et doivent être données.

Les noeuds du graphe de précedence forment l'ensemble suivant:

$N = \{début, fin\} \cup C^j$, avec:

- *début*, un noeud qui marque l'origine temporelle du processus d'assemblage,
- *fin*, un noeud qui marque le terme temporel du processus d'assemblage,
- C^j , l'ensemble des caractéristiques de p^j tel que décrit au point précédent.

Un arc du graphe est une paire ordonnée dont le premier élément est un noeud n^1 et le second élément est un noeud n^2 , tel que n^1 doit être établi *avant* n^2 . L'absence d'arc entre deux noeuds signifie que l'établissement de l'un ou l'autre peut se faire indifféremment avant ou après, voire en même temps.

Le graphe de précedence a les propriétés suivantes:

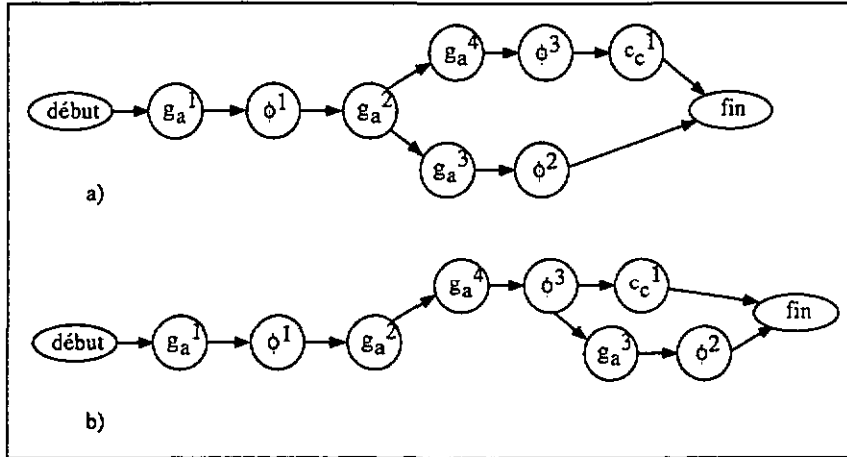


Figure 7.2: Graphes de précédence possibles pour la description de l'espace fonctionnel retenu pour la valve de vélo

- $\forall n^1, n^2, n^3 \in N$, si il existe la succession d'arcs $(n^1, n^2)(n^2, n^3)$ alors cette succession est un chemin entre n^1 et n^3 , avec pour conséquence que n^1 doit être établi avant n^3 .
- $\forall n^i \in N$, il existe un chemin tel que (n^i, fin)
- $\forall n^i \in N$, il existe un chemin tel que $(début, n^i)$

7.2.1 Exemple

La figure 7.2 donne deux exemples de graphes de précédence possibles pour la valve de vélo. Le graphe b) indique que l'on ne s'intéresse qu'aux processus d'assemblage qui réalisent d'abord l'assemblage de l'écrou à la soupape, puis l'assemblage du guide au corps. Ces processus d'assemblage sont également compris dans la description faite par le graphe a), mais celui-ci inclut également les processus qui font l'inverse.

7.3 Les processus d'assemblage

Un *processus d'assemblage* est l'énumération de la succession des caractéristiques du produit qui amène des pièces individuelles au produit complètement assemblé. Elle représente un ordre partiel pour la réalisation des caractéristiques qui la composent.

Le graphe ET/OU, noté G_{AV} , permet de représenter tous les processus d'assemblage retenus pour le système, y compris les constituants (sous-assemblages) impliqués par chacun d'eux.

L'ensemble des noeuds du graphe ET/OU d'un produit correspond de façon bi-univoque à l'ensemble Σ des sous-assemblages.

Un sous-assemblage Σ^k est décrit par le couple:

$\Sigma^k = \langle P^k, C^k \rangle$, avec:

- P^k , l'ensemble des composants élémentaires du sous-assemblage. $P^k = (c_e^1, c_e^2, \dots, c_e^m)$ avec m , le nombre de composants élémentaires dans le sous-assemblage.
- C^k , l'ensemble des caractéristiques déjà réalisées dans le sous-assemblage.

Les hyperarcs correspondent aux tâches d'assemblage géométriquement et technologiquement faisables, c'est-à-dire dans notre cas, la réalisation des caractéristiques du produit:

- chaque hyperarc correspondant à une *tâche géométrique* (réalisation d'une caractéristique géométrique d'assemblage g_a) est une paire ordonnée, dont le premier élément est le noeud représentant un sous-assemblage Σ^k , et le second un ensemble de deux noeuds $\{\Sigma^i, \Sigma^j\}$ tel que $\Sigma^i \cup \Sigma^j = \Sigma^k$ soit $P^i \cup P^j = P^k$ et $C^i \cup C^j \cup \{g_a\} = C^k$
- un hyperarc correspondant à une *tâche physique* (réalisation d'une caractéristique physique ϕ) ou à une *tâche complémentaire* (réalisation d'une caractéristique complémentaire c_c) est une paire ordonnée, dont le premier élément est un noeud représentant un sous-assemblage Σ^k , et le second un noeud Σ^i tel que $P^k = P^i$ et $C^i \cup \{\phi \text{ (ou } c_c)\} = C^k$.

Tout sous-graphe de G_{AV} ne comportant que des hyperarcs ET et ayant:

- pour noeud initial Σ^0 (voir note ² ci-dessous), et
- pour noeuds finaux $(c_e^1, \{\emptyset\}), (c_e^2, \{\emptyset\}), \dots, (c_e^n, \{\emptyset\})$, n étant le nombre de composants élémentaires du produit.

est un arbre représentant un processus d'assemblage exécutable.

Un tel arbre introduit un ordre partiel de ses hyperarcs: un hyperarc h^i est considéré comme *précédent* un hyperarc h^j dans l'arbre, si h^i *quitte* le noeud n^k et h^j *arrive* sur n^k . Par définition, toute séquence d'hyperarcs correspondant à cet ordre partiel, forme un processus d'assemblage du produit.

7.3.1 Détermination du graphe ET/OU

Le graphe ET/OU d'un produit se déduit de l'arbre des références opératoires A_{ro} et du graphe de précedence G_p . Il se construit par décomposition du noeud Σ^0 jusqu'à n'avoir que le produit totalement désassemblé. La décomposition de chaque noeud se fait en conformité avec le graphe de précedence, et A_{ro} .

L'algorithme de décomposition d'un noeud Σ^k est le suivant:

- Σ^k définit un sous-arbre de A_{ro} et un sous-graphe de G_p . Ces deux sous-graphes sont définis par:

$$- A_{ro}(\Sigma^k) = \langle P, L \rangle \text{ avec}$$

$$* P = \{e \mid [e = e^x] \wedge [e^x \in (A_{ro} \cap \Sigma^k)]\}, \text{ l'ensemble des pièces et des caractéristiques communes à } A_{ro} \text{ et } \Sigma^k.$$

² Σ^0 représente le produit fini totalement assemblé, c'est-à-dire que Σ^0 contient les n pièces du produit p^j et que toutes les caractéristiques ont été réalisées. $\Sigma^0 = (P^j, C^j)$

* $L = \{(e^1, e^2) \mid [e^1, e^2 \in \Sigma^k] \wedge [\exists(e^1, e^2) \in R_{rg}]\}$, l'ensemble des relations existantes entre les éléments de P^k et de C^k qui constituent Σ^k .

- $G_p(\Sigma^k) = \langle N_p, P_r \rangle$ avec

* $N_p = \{C^k \cup \{début, fin\}\}$, l'ensemble des caractéristiques existantes dans Σ^k auquel sont ajoutés les noeuds *début* et *fin* de G_p .

* $P_r = \{(c^1, c^2) \mid [c^1, c^2 \in N_p] \wedge [\exists(c^1, c^2) \in G_p]\}$, l'ensemble des relations de précédence existant entre les noeuds de N_p .

Remarque: le produit fini Σ^0 a, pour sous-arbre, l'arbre des références opératoires complet ($A_{ro}(\Sigma^0) = A_{ro}$), et pour sous-graphe de précédence, le graphe de précédence complet ($G_p(\Sigma^0) = G_p$).

- Faire la liste $C_{exec}(\Sigma^k)$ des caractéristiques exécutables depuis Σ^k :
 $C_{exec}(\Sigma^k) = \{c \mid \exists(c, fin) \in G_p(\Sigma^k)\}$. On ne retient que les caractéristiques dont la réalisation se fait le plus tard possible.
- Chaque $c \in C_{exec}(\Sigma^k)$ définit un nouvel hyperarc h du graphe ET/OU tel que:
 - Si c est une caractéristique géométrique d'assemblage:
 $h = (\Sigma^k, \{\Sigma^i, \Sigma^j\})$ avec
 - * $\Sigma^i = \{e \mid [e \in A_{ro}(\Sigma^k)] \wedge [\exists \text{ un chemin ds } (A_{ro}(\Sigma^k) - \{c\}) \text{ de } e \text{ au composant de } c]\}$
 - * $\Sigma^j = \{e \mid [e \in A_{ro}(\Sigma^k)] \wedge [\exists \text{ un chemin ds } (A_{ro}(\Sigma^k) - \{c\}) \text{ de } e \text{ au récepteur de } c]\}$
 - Si c est une caractéristique physique ou une caractéristique complémentaire: $h = (\Sigma^k, \Sigma^i)$ avec $P^k = P_i$ et $C_i \cup \{c\} = C^k$
- chaque hyperarc généré est ajouté à G_{AV} , ainsi que chaque sous-assemblage Σ généré qui n'est pas déjà $\in G_{AV}$.

7.3.2 Exemple

Le graphe ET/OU représenté à la figure 7.3 a été construit selon l'algorithme donné ci-dessus, par la décomposition de l'arbre des références opératoires (figure 7.1), en contrôlant les contraintes de précédence données dans le graphe a) de la figure 7.2.

7.4 Le découpage fonctionnel

La description des processus d'assemblage se fait à différents niveaux d'abstraction. Les niveaux fonctionnels utilisés dans AMACOLA, selon les niveaux décrits au point 5.3.1, sont: le niveau des tâches d'assemblage, le niveau des opérations d'assemblage et le niveau des fonctions élémentaires.

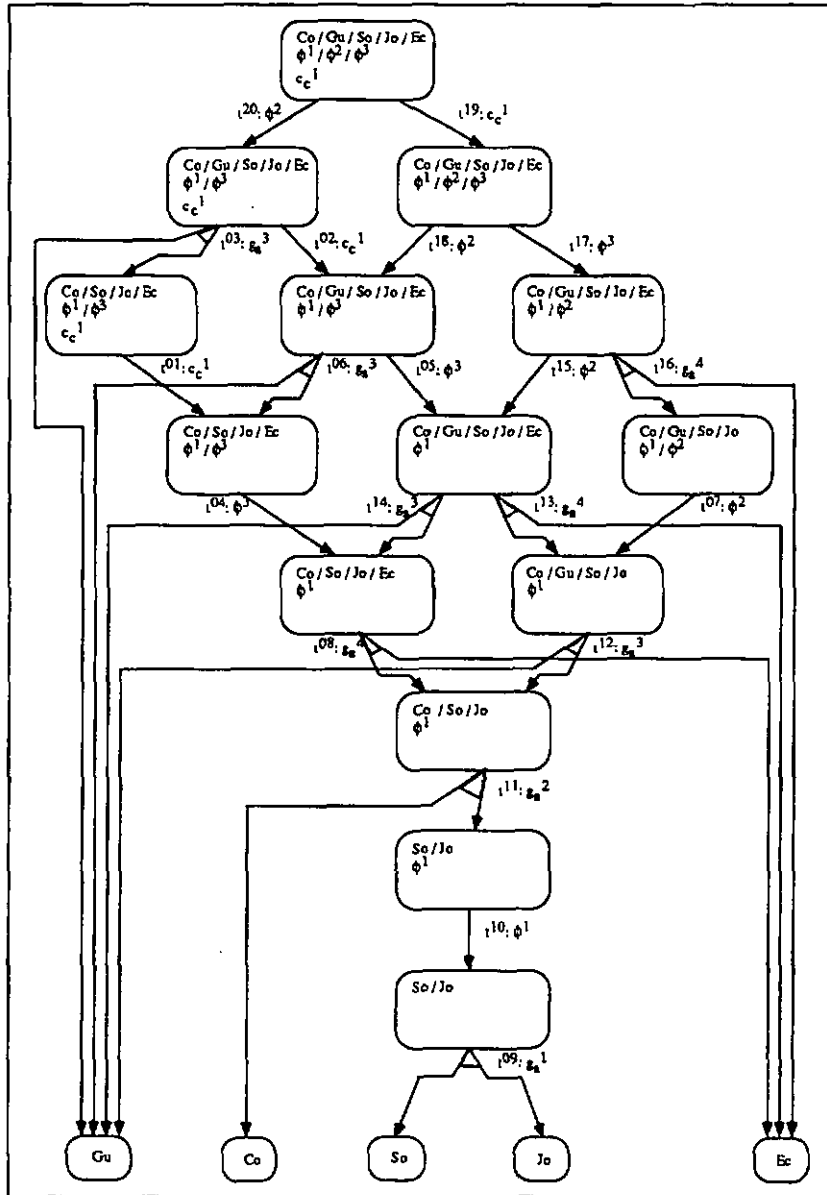


Figure 7.3: Graphe ET/OU représentant les processus d'assemblage de l'espace fonctionnel retenu pour la valve de vélo

7.4.1 Les tâches d'assemblage

Une *tâche d'assemblage* est la création d'une caractéristique du produit. Les tâches d'assemblage sont générées au moment de la création de G_{AV} , chacun de ses hyperarcs représentant une tâche d'assemblage. Une tâche d'assemblage t^k est décrite par le quintuplet:

$t^k = \langle c^k, \Sigma_r^k, \Sigma_c^k, \Sigma^k, O^k \rangle$ avec:

- c^k , la caractéristique que réalise t^k ,
- Σ_r^k , le sous-assemblage récepteur,
- Σ_c^k , le sous-assemblage composant. Si c^k est une caractéristique physique ou une caractéristique complémentaire, $\Sigma_c^k = \{\emptyset\}$,
- Σ^k , le sous-assemblage résultant de la réalisation de c^k ,
- O^k , l'ensemble des opérations d'assemblage qui constituent t^k . Au départ, $O^k = \{\emptyset\}$.

On distingue trois types de tâches d'assemblage, selon les caractéristiques qu'elles réalisent: les tâches géométriques, les tâches physiques et les tâches complémentaires.

7.4.2 Les opérations d'assemblage

Une *opération d'assemblage* est une modification de l'état géométrique d'un sous-assemblage donné (position, orientation, nombre de pièces constitutives, degrés de liberté). Le découpage des tâches d'assemblage en opérations d'assemblage peut se faire de deux manières:

- de manière automatique: AMACOIA applique un découpage fonctionnel correspondant à des solutions courantes dans le milieu industriel. Exemples: le découpage d'une tâche géométrique détermine 3 opérations d'assemblage: la localisation du récepteur, la distribution du composant et le positionnement du composant sur le récepteur. Le découpage d'une tâche physique détermine 2 opérations d'assemblage: la localisation du récepteur et l'attachement.
- de manière interactive: dans les cas particuliers, l'utilisateur peut décrire le découpage fonctionnel de la tâche d'assemblage. Ceci s'applique principalement aux tâches complémentaires.

On distingue cinq types d'opérations d'assemblage: distribution, localisation, positionnement, attachement et opération complémentaire. Une opération d'assemblage o^k est décrite par le quintuplet:

$o^k = \langle goal^k, \Sigma^k, e_i^k, e_o^k, F^k \rangle$ avec:

- $goal^k$, le type de l'opération d'assemblage: distribution, localisation, positionnement, attachement ou opération complémentaire,
- Σ^k , le sous-assemblage sur lequel porte o^k ,

- e_i^k , l'état géométrique de Σ^k avant l'exécution de l'opération. Un état géométrique est décrit par le sous-assemblage et son orientation par rapport au repère absolu,
- e_o^k , l'état géométrique de Σ^k après l'exécution de l'opération. Il est décrit par le sous-assemblage résultant et son orientation par rapport au repère absolu,
- F^k , l'ensemble des fonctions élémentaires qui constituent σ^k . Au départ, $F^k = \{\emptyset\}$.

7.4.3 Les fonctions élémentaires

Une *fonction élémentaire* est une modification élémentaire de l'état géométrique d'un sous-assemblage. Nous considérons trois types de fonctions élémentaires: mouvement, force et localisation. Le découpage des opérations d'assemblage en fonctions élémentaires peut se faire de la manière suivante:

- pour chaque face accessible du sous-assemblage concerné, AMACOIA détermine une fonction élémentaire de localisation, le choix des fonctions qui seront véritablement réalisées se faisant au moment de la conception des équipements.
- pour les opérations de positionnement, AMACOIA détermine les fonctions de mouvement nécessaire sur la base de e_i^k et de e_o^k , par comparaison des orientations des deux états.
- pour les opérations d'attachement et de positionnement (saisie), AMACOIA détermine si nécessaire les fonctions élémentaires de force.

Comme pour le découpage des tâches d'assemblage, le découpage des opérations d'assemblage en fonctions élémentaires peut aussi se faire de manière interactive.

7.5 Conclusion

Le module d'initialisation a pour objectifs la saisie du sous-espace de l'espace fonctionnel de la conception des systèmes d'assemblage que souhaite explorer l'utilisateur, et la transformation de ce sous-espace sous une forme utilisable par les autres modules d'AMACOIA.

Pour décrire l'espace de recherche retenu, l'utilisateur entre l'arbre des références opératoires et un graphe de précedence:

- l'arbre des références opératoires est un modèle opératoire du produit. Il permet de décrire ses composants élémentaires et ses caractéristiques (géométriques, physiques, complémentaires) à réaliser au cours de l'assemblage, ainsi que certains choix technologiques pour leur réalisation (choix des références et des directions de travail).
- le graphe de précedence permet de décrire les contraintes de précedence pour la réalisation des caractéristiques, représentant les solutions aux contraintes-produit ou découlant des critères stratégiques de production retenus par l'utilisateur.

Ainsi, l'espace de recherche fonctionnel du système AMACOIA, n'est composé que de processus d'assemblage admissibles pour le produit qui satisfont aux contraintes-produit.

Le module d'initialisation transforme alors cette description en un graphe ET/OU qui regroupe les tâches d'assemblage et les constituants impliqués, formant les processus d'assemblage retenus. Le but de cette transformation est de permettre l'évaluation et la sélection des processus d'assemblage par le module de sélection.

La description des tâches d'assemblage formant le graphe ET/OU est ensuite précisée par le découpage fonctionnel en opérations, qui sont elles-mêmes découpées en fonctions élémentaires. Le but de ce découpage fonctionnel est d'initialiser les données manipulées par les agents du module de conception.

Ainsi, le résultat produit par le module de sélection est l'expression de l'espace fonctionnel retenu par l'utilisateur sous la forme d'un graphe ET/OU. Chacune des tâches d'assemblage formant les processus est exprimée à trois niveaux d'abstraction: le niveau 1 des tâches et constituants, le niveau 2 des opérations et constituants localisés, et le niveau 3 des fonctions élémentaires et constituants localisés.

Chapitre 8

Le module de sélection

Le module de sélection traite l'espace fonctionnel de la conception des systèmes d'assemblage. Son objectif est de déterminer, parmi les processus d'assemblage, celui dont les tâches sont les plus simples à réaliser (d'un point de vue fonctionnel). Le parcours de l'espace fonctionnel à la recherche de son optimum se fait de manière itérative. Une itération consiste:

1. à sélectionner un processus d'assemblage admissible constituant un point de l'espace fonctionnel. En d'autres termes, l'extraction d'un point parmi les points constituant l'espace fonctionnel.
2. à évaluer le processus d'assemblage obtenu. Selon le résultat de cette évaluation, le processus d'assemblage devient ou non le nouveau processus courant. Le processus d'assemblage courant devient alors le point de départ d'une nouvelle itération.

Le module de sélection ne travaille pas sur l'espace fonctionnel complet de la conception des systèmes d'assemblage, constitué de tous les processus d'assemblage possibles du produit. Son espace de recherche est un sous-espace de l'espace des solutions de l'espace fonctionnel, constitué des processus d'assemblage admissibles qui satisfont aux contraintes-produit. Ce sous-espace est fourni au module de sélection par le module d'initialisation: c'est le graphe ET/OU, G_{AV} , généré par le module d'initialisation et qui regroupe l'ensemble des processus d'assemblage admissibles retenus par l'utilisateur.

Le résultat fourni par le module de sélection est un processus d'assemblage (optimal ou non) qui sera utilisé par le module de conception d'AMACOIA pour déterminer les flots d'assemblage.

Le module de sélection est un système multi-agents, basé sur le modèle de l'éco-résolution. La sélection d'un processus d'assemblage de G_{AV} est obtenue par la dynamique du système multi-agents, alors que l'évaluation du processus n'est effectuée que par un seul agent.

8.1 La population d'agents

Les agents qui composent le SMA de sélection sont définis principalement pour résoudre les problèmes inhérents à la sélection d'un processus d'assemblage du graphe ET/OU. G_{AV} .

Le graphe ET/OU regroupe toutes les tâches constitutives qui peuvent être réalisées pour assembler le produit, et fixe un ordre partiel pour leur exécution (arcs ET), ainsi que les différentes possibilités qu'elles représentent (arcs OU). Un processus d'assemblage est une succession de tâches, amenant du pré-produit au produit fini, qui respecte l'ordre partiel fixé par G_{AV} , et où toute alternative a disparu. Sélectionner un processus d'assemblage revient dès lors à combiner des tâches du graphe ET/OU de manière à respecter cette définition du processus d'assemblage.

Ainsi la population d'agents A_{ms} du module de sélection est définie par $A_{ms} = A_t \cup \{a_p\}$ avec:

- A_t , l'ensemble des agents-tâches, générés sur la base des tâches de G_{AV} . Les agents-tâches vont chercher à se combiner, au travers de l'agent-processus, pour former un processus d'assemblage.
- a_p , l'agent-processus. Il est chargé de vérifier la cohérence de la combinaison des agents-tâches, par rapport à la définition du processus d'assemblage. C'est également l'agent-processus qui gère le parcours de l'espace de recherche.

Le lecteur trouvera une description détaillée (objectifs, connaissances, comportements) des agents du module de sélection dans l'annexe A.

Deux autres agents sont encore associés à cette population: l'utilisateur du système, a_u , et l'agent-flot, a_i , du module de conception.

A_{ms} constitue l'environnement global des agents du SMA de sélection. Chaque agent n'en a qu'une vue partielle en fonction de ses objectifs et des compétences qui lui sont attribuées.

8.2 Dynamique du SMA de sélection

Selon le modèle de l'éco-résolution, le comportement des agents du module de sélection est guidé par la recherche de satisfaction: la réception d'un message en provenance d'une accointance a pour effet de modifier l'état de satisfaction de l'agent, et il déclenche alors une réaction appropriée à la modification, afin de se rapprocher de ses conditions de satisfaction.

Les conditions de satisfaction des agents du module de sélection portent sur la cohérence d'une combinaison de tâches, par rapport à la définition du processus d'assemblage:

- un agent-tâche est satisfait s'il fait partie d'une combinaison de tâches cohérente. Autrement dit, s'il fait partie du processus d'assemblage sélectionné.
- l'agent-processus est satisfait si la combinaison de tâches constitue un processus d'assemblage complet qui, selon la définition, amène du pré-produit au produit fini.

La population d'agents est engendrée par le système: l'agent-processus et un agent-tâche par tâche constitutive contenue dans le graphe ET/OU.

8.2.1 Sélection d'un processus d'assemblage

La sélection d'un processus d'assemblage est obtenue par la dynamique suivante: chaque agent-tâche, une fois généré, va chercher à remplir sa condition de satisfaction en envoyant une requête de placement à l'agent-processus.

L'agent-processus traite les requêtes des agents-tâches dans l'ordre de leurs arrivées. Selon que la tâche de l'agent-tâche requérant peut être intégrée ou non au processus partiel en construction (en respect avec l'ordre partiel de G_{AV} et en refusant les alternatives), l'agent-processus accepte ou rejette la requête, et envoie la réponse.

Selon cette réponse, l'agent-tâche envoie une nouvelle requête de placement à l'agent-processus si sa requête a été rejetée, sinon il ne fait rien, sa condition de satisfaction étant remplie.

Chaque intégration d'un agent-tâche par l'agent-processus le rapproche de sa condition de satisfaction, et le processus d'assemblage se construit au fur et à mesure des acceptations, à partir de la première de celles-ci, jusqu'à l'obtention d'un processus d'assemblage complet et la satisfaction de l'agent-processus.

Ainsi, un processus d'assemblage est sélectionné par la dynamique que composent les requêtes, acceptations et rejets, et dont la limite est marquée par la satisfaction de l'agent-processus.

8.2.2 Parcours de l'espace

Le parcours de l'espace de recherche est géré uniquement par l'agent-processus. Une fois que le processus d'assemblage, construit selon la dynamique décrite ci-dessus, est complet (et donc que l'agent-processus est satisfait), il applique le recuit simulé. Sur la base de cette évaluation, l'agent-processus accepte ou non alors le nouveau processus sélectionné comme le nouveau processus courant. Puis, sur la base du processus courant (qu'il ait changé ou non), avec pour conséquence la perte de sa satisfaction, il commence à construire un nouveau processus d'assemblage en continuant de traiter les requêtes qu'il reçoit des agents-tâches insatisfaits. La trajectoire du parcours de l'espace de recherche est ainsi constituée des processus courants successifs de l'agent-processus.

L'optimum recherché est le processus d'assemblage dont les tâches sont les plus simples à réaliser. L'évaluation de la difficulté d'exécution des tâches d'assemblage se base sur les critères fonctionnels de stabilité, d'accessibilité, de parallélisation et d'utilisation des récepteurs. Ainsi, la fonction minimisée est:

$$f_d(p) = \sum_{k=1}^n d_t^k \quad (8.1)$$

- p , le processus d'assemblage
- n , le nombre de tâches d'assemblage dans le processus d'assemblage p
- d_t^k , la difficulté estimée d'exécution de la tâche constitutive k .

$$d_t^k = \frac{1}{4}(\sigma^k + \alpha^k + \pi^k + \rho^k) \text{ avec:}$$

- σ^k , le facteur de stabilité de la tâche d'assemblage k . C'est une valeur, comprise entre 1 et 10, calculée sur la base des degrés de libertés bloqués du récepteur de la caractéristique que réalise la tâche.

- α^k , le facteur d'accessibilité de la tâche d'assemblage k . C'est une valeur, comprise entre 1 et 10, calculée sur la base des obstacles pouvant occuper le volume délimité par la face du récepteur de la caractéristique de la tâche, et s'étendant selon la direction de travail.
- π^k , le facteur de parallélisation de la tâche d'assemblage k . C'est une valeur, comprise entre 1 et 10, calculée sur la base du rapport entre les longueurs des deux branches du processus d'assemblage qui peuvent être exécutées en parallèle jusqu'à la tâche k .
- ρ^k , le facteur d'utilisation des récepteurs de la tâche d'assemblage k . C'est une valeur, comprise entre 1 et 10, calculée sur la base des récepteurs des caractéristiques des tâches amont et du récepteur de la caractéristique de la tâche k , ainsi que des directions de travail¹.

Chaque fois que l'agent-processus applique le recuit simulé, la probabilité d'acceptation du nouveau processus est:

$$a(T^{\alpha_p}) = \begin{cases} 1 & \text{si } f_d(p_{new}^{\alpha_p}) \leq f_d(p_{crt}^{\alpha_p}) \\ \exp\left(\frac{f_d(p_{new}^{\alpha_p}) - f_d(p_{crt}^{\alpha_p})}{T^{\alpha_p}}\right) & \text{si } f_d(p_{new}^{\alpha_p}) > f_d(p_{crt}^{\alpha_p}) \end{cases}$$

avec:

- T^{α_p} , la valeur de la tolérance au changement (température) de l'agent-processus.
- $p_{crt}^{\alpha_p}$, le processus d'assemblage courant,
- $p_{new}^{\alpha_p}$, le processus d'assemblage qui vient d'être sélectionné

Si le processus sélectionné est accepté, il devient alors le nouveau processus courant. En cas d'acceptation de $p_{new}^{\alpha_p}$ dans le cas où $f_d(p_{new}^{\alpha_p}) > f_d(p_{crt}^{\alpha_p})$, l'agent-processus diminue sa tolérance au changement.

Le parcours de l'espace s'arrête lorsque, au moment de l'application du recuit simulé, la tolérance au changement T^{α_p} de l'agent-processus est nulle. Tous les agents-tâches encore insatisfaits cherchant à se placer après cet état de l'agent-processus sont alors désactivés.

Ainsi, le critère de terminaison (selon la définition donnée au point 2.6) indiquant qu'un parcours de l'espace de recherche est achevé, est que l'agent-processus est satisfait ($p_{new}^{\alpha_p}$ est un processus d'assemblage complet), que sa tolérance au changement est nulle, et que tous les agents-tâches sont soit satisfaits, soit désactivés.

Un nouveau parcours de l'espace fonctionnel est alors possible (et donc une nouvelle trajectoire ne donnant pas forcément les mêmes résultats), sur demande de l'utilisateur ou de l'agent-flot du module de conception, par l'envoi à l'agent-processus d'un message d'activation. Ce message aura pour effet de réinitialiser la tolérance au changement de l'agent-processus et de réactiver les agents-tâches.

A tout moment du parcours de l'espace de recherche, l'agent-processus peut fournir au module de conception son processus d'assemblage courant ou son processus optimal du moment, sur simple demande de l'utilisateur ou de l'agent-flot.

¹Pour le détail des calculs des différents facteurs, voir l'annexe C

8.3 Conclusion

Les objectifs du module de sélection sont liés au parcours de l'espace recherche, sous-espace de l'espace fonctionnel de la conception des systèmes d'assemblage, donné sous la forme du graphe ET/OU, G_{AV} , généré par le module d'initialisation. L'optimum recherché est le processus d'assemblage dont les tâches sont les plus simples à réaliser (d'un point de vue fonctionnel).

Le parcours de l'espace de recherche est obtenu par la dynamique des agents constituant le module de sélection. Ce parcours est un phénomène itératif, dont une étape consiste:

1. à sélectionner un processus d'assemblage de G_{AV} . Ceci est obtenu par l'interaction entre les agents-tâches et l'agent-processus: chaque agent cherche à remplir ses conditions de satisfaction, les agents-tâches en cherchant à faire partie du processus sélectionné, l'agent-processus en maintenant la cohérence de la combinaison de tâches, jusqu'à l'obtention d'un processus d'assemblage complet. L'échange des requêtes et des réponses entre les agents-tâches et l'agent-processus se poursuit jusqu'à la satisfaction de ce dernier. Le processus d'assemblage est sélectionné.
2. à évaluer le processus d'assemblage obtenu. Cette évaluation est faite par l'agent-processus en appliquant le recuit simulé. Selon le résultat de l'évaluation, le processus sélectionné devient ou non le nouveau processus courant. Le processus d'assemblage courant constitue alors le point de départ d'une nouvelle étape du parcours.

Le parcours de l'espace de recherche s'arrête lorsque l'agent-processus est satisfait, sa tolérance au changement est nulle, et que tous les agents-tâches sont soit satisfaits, soit désactivés.

Les résultats du module de sélection sont des processus d'assemblage (optimaux ou non, selon que l'agent-processus fournisse son processus optimal du moment ou son processus courant) qui sont donnés en entrée du module de conception.

Chapitre 9

Le module de conception

Le module de conception traite l'espace physique de la conception des systèmes d'assemblage. Son objectif est de déterminer, parmi les flots d'assemblage, l'ilot d'assemblage le moins cher. Comme pour le module de sélection, le parcours de l'espace physique à la recherche de son optimum se fait de manière itérative. Une itération consiste:

1. à déterminer un flot d'assemblage qui satisfait aux contraintes-ressources (temps de cycle contrat, fiabilité, flexibilité). En d'autres termes, à générer un point de l'espace des solutions de l'espace physique.
2. à évaluer le coût de l'ilot d'assemblage obtenu. Selon le résultat de cette évaluation, l'ilot d'assemblage devient ou non le nouvel ilot courant. L'ilot d'assemblage courant devient alors le point de départ d'une nouvelle itération.

La détermination des flots d'assemblage est basée sur un *processus opératoire*¹. Un processus opératoire regroupe non seulement les tâches constitutives (géométriques, physiques et complémentaires), mais aussi les tâches positionnelles (réorientations, chargements, déchargements) que devra réaliser l'ilot d'assemblage. Les processus opératoires utilisés par le module de conception sont obtenus par transformation des processus d'assemblage fournis par le module de sélection.

Le résultat fourni par le module de conception est une spécification d'un ilot d'assemblage (optimal ou non) qui décrit ses équipements constitutifs, leur organisation en postes et cellules, son temps de cycle et son coût.

Le module de conception est un système multi-agents. Comme pour le module de sélection, ce SMA est basé sur le modèle de l'éco-résolution. La génération d'un ilot d'assemblage est obtenue par la dynamique du système, alors que l'évaluation des ilots n'est effectuée que par un seul agent.

9.1 La population d'agents

Les agents qui composent le SMA de conception sont définis principalement pour gérer les problèmes inhérents à la spécification d'un ilot d'assemblage.

¹ cf. §5.8

La spécification d'un flot d'assemblage comporte les deux aspects de la détermination de ses équipements constitutifs, et de leur organisation en son sein:

- dans AMACOLA, la détermination des équipements constitutifs est basée sur la notion de *ressource générique*. Une ressource générique est une spécification fonctionnelle d'un type d'équipement: elle est définie par une fonction (maintenir, transférer, etc.), les constituants impliqués et des points de travail. Les ressources génériques permettent en fait de décrire les modes opératoires du processus opératoire. Elles sont ensuite matérialisées par les équipements de l'ilot d'assemblage.
- l'organisation des équipements constitutifs au sein de l'ilot d'assemblage consiste dans la détermination des postes et des cellules d'assemblage de l'ilot.

Ainsi la population d'agents A_{mc} du module de conception est définie par $A_{mc} = A_t^{pop} \cup A_p \cup A_c \cup \{a_i\}$ avec:

- A_t^{pop} , l'ensemble des agents-tâches² générés sur la base des tâches du processus opératoire p_{op} . Les agents-tâches sont chargés de créer les ressources génériques.
- A_p , l'ensemble des agents-postes chargés de créer les postes de l'ilot d'assemblage sur la base des ressources génériques des agents-tâches.
- A_c , l'ensemble des agents-cellules chargés de constituer les cellules de l'ilot d'assemblage sur la base des postes créés par les agents-postes.
- a_i , l'agent-ilot chargé de constituer l'ilot d'assemblage proprement dit sur la base des cellules créées par les agents-cellules. C'est également l'agent-ilot qui gère l'exploration de l'espace de recherche.

Le lecteur trouvera une description détaillée (objectifs, connaissances, comportements) des agents du module de conception dans l'annexe B.

La détermination des équipements constitutifs est le fait des agents-tâches et des agents-postes. L'organisation des équipements est le fait des agents-postes, des agents-cellules et de l'agent-ilot.

L'utilisateur du système, a_u , et l'agent-processus, a_p , du module de sélection sont également associés à cette population.

A_{mc} constitue l'environnement global des agents du SMA de conception. Comme dans le module de sélection, chaque agent n'en a qu'une vue partielle en fonction de ses objectifs et des compétences qui lui sont attribuées.

9.2 Dynamique du SMA de conception

Comme dans le module sélection, le comportement des agents du module de conception est guidé par la recherche de satisfaction: la réception d'un message en provenance d'une accointance modifie l'état de satisfaction de l'agent, et il

²Remarque: les agents-tâches du module de conception sont différents de ceux du module de sélection.

déclenche alors une réaction appropriée à la modification, afin de se rapprocher de ses conditions de satisfaction.

Les conditions de satisfaction des agents du module de conception portent sur des critères technologiques et sur des critères d'intégration:

- la condition de satisfaction technologique est liée à l'élaboration de la structure dont est responsable l'agent (ressources génériques, poste, cellule, flot). Une fois que la condition technologique est remplie, l'agent refuse toute nouvelle requête en rapport avec elle, qui pourrait survenir. Par contre, il s'adapte encore aux modifications dynamiques des éléments de sa structure, avec parfois pour conséquence, la perte de sa satisfaction technologique.
- la condition de satisfaction d'intégration est liée au fait que les structures créées par les agents sont construites par intégration d'éléments fournis par les autres agents: un poste est construit sur un ensemble de ressources génériques, une cellule sur un ensemble de postes et l'flot sur un ensemble de cellules. La condition d'intégration est remplie lorsque l'agent a été intégré, en tant qu'élément, dans la structure adéquate: un agent-tâche dans un agent-poste, un agent-poste dans un agent-cellule et un agent-cellule dans l'agent-flot. Une fois que la condition d'intégration est remplie, l'agent ne fait rien.

La condition de satisfaction d'intégration pousse l'agent à envoyer une requête de placement à l'une de ses accointances susceptible de l'intégrer, dont la condition de satisfaction technologique lui permettra alors de décider si elle accepte ou non cette requête. Le tableau 9.1 résume les conditions de satisfaction des agents du module de conception.

A l'exception de l'agent-flot, généré par le système, tous les agents du module de conception sont générés dynamiquement:

- les agents-tâches par l'agent-flot, sur la base du processus opératoire;
- les agents-postes par les agents-tâches, au fur et à mesure des besoins en nouveaux postes (le premier agent-poste est créé par le premier agent-tâche activé);
- les agents-cellules par les agents-postes, au fur et à mesure des besoins en nouvelles cellules (le premier agent-cellule est créé par le premier agent-poste);

La vie d'un agent est liée à sa condition d'existence: si, à un moment donné, cette condition n'est plus remplie, l'agent envoie un avis de décès à ses accointances et disparaît. Les conditions d'existence des agents du module de conception sont:

- pour les agents-tâches: la tâche pour laquelle l'agent a été créé fait toujours partie du processus opératoire;
- pour les agents-postes: il y a toujours au moins un agent-tâche placé sur l'agent-poste;
- pour les agents-cellules: il y a toujours au moins un agent-poste placé sur l'agent-cellule;

L'agent-flot est immortel, il n'a donc pas de condition d'existence.

	Cond. technologique	Cond. d'intégration	Objectifs
Agents-tâches	Les ressources génériques ont été créées	Les ressources génériques sont intégrées dans un poste	Création des ressources génériques et leur intégration dans un poste
Agents-postes	La différence entre temps de cycle contrat et temps de cycle du poste est minimale	Le poste est intégré dans une cellule	Intégration des ressources génériques, transformation en équipement et intégration dans une cellule
Agents-cellules	La cellule constitue l'ensemble cohérent de postes le plus grand possible	La cellule est intégrée dans l'îlot	Intégration des postes, détermination des transferts et intégration dans l'îlot
Agent-îlot	L'îlot construit couvre l'ensemble du processus opératoire	-	Intégration des cellules, détermination des transferts et application du recuit

Tableau 9.1: Résumé des conditions de satisfaction et des objectifs des agents du module de conception

9.2.1 Détermination d'un îlot d'assemblage

La génération d'un îlot d'assemblage est obtenue par la dynamique suivante: l'agent-îlot commence par transformer en processus opératoire le processus d'assemblage fourni par le module de sélection. Il génère ensuite les agents-tâches, à raison d'un agent-tâche par tâche du processus opératoire, et les active.

Chaque agent-tâche génère alors ses ressources génériques et satisfait ainsi à sa condition technologique. Puis, poussé par sa condition d'intégration, il envoie une requête de placement à un agent-poste.

Un agent-poste traite les requêtes des agents-tâches dans l'ordre de leurs arrivées. Chaque intégration, en respect avec les contraintes de précédence, le rapproche de sa condition technologique (la différence entre le temps de cycle du poste créé et le temps de cycle contrat est minimale). L'agent-poste crée son poste en regroupant les ressources génériques et en matérialisant ces regroupements dans des équipements physiques (robots, pinces, posages, etc.). Le poste se construit ainsi, au fur et à mesure des acceptations, jusqu'à ce que la condition technologique soit remplie. A partir de là, l'agent-poste refuse toute nouvelle requête (mais réagit toujours aux avis de décès des agents-tâches déjà placés), ce qui aura pour conséquence la création de nouveaux agents-postes par les agents-tâches encore insatisfaits. Si sa condition d'intégration n'est pas remplie, l'agent-poste envoie une requête de placement à un agent-cellule.

Chaque fois qu'un agent-poste a modifié son poste à la suite de l'intégration d'un nouvel agent-tâche, il en avertit l'agent-cellule sur lequel il est placé, afin que ce dernier puisse tenir compte de ces modifications.

Comme les agents-postes, un agent-cellule traite les requêtes des agents-postes dans l'ordre de leurs arrivées. De même, chaque intégration, en respect avec les contraintes de précédence, le rapproche de sa condition technologique (la cellule créée doit constituer l'ensemble cohérent (technologiquement) de postes, le plus grand possible). Par contre, la modification d'un agent-poste, qui a pour conséquence le rejet par l'agent-cellule d'agents-postes déjà placés, l'en éloigne, et peut même conduire au suicide de l'agent-cellule, si sa condition d'existence n'est plus remplie. La cellule est construite en appliquant des règles de construction qui garantissent la cohérence. Dès que la condition technologique est atteinte, l'agent-cellule refuse toute nouvelle requête (mais réagit toujours aux messages de modification), ce qui aura pour conséquence la création de nouveaux agents-cellules par les agents-postes encore insatisfaits. Si sa condition d'intégration n'est pas remplie, l'agent-cellule envoie une requête de placement à l'agent-flot.

Chaque fois qu'un agent-cellule a modifié sa cellule à la suite de l'intégration ou de la modification d'un agent-poste, il en avertit l'agent-flot, afin que ce dernier puisse tenir compte de ces modifications.

L'agent-flot accepte automatiquement toutes les requêtes des agents-cellules. L'agent-flot, en réaction aux requêtes, aux messages de modification et aux avis de décès des agents-cellules, ne fait que contrôler sa condition technologique (les cellules doivent couvrir la totalité du processus opératoire). Une fois que sa condition technologique est remplie (ce qui implique aussi que tous les autres agents sont satisfaits), il choisit les équipements de transfert entre les cellules, et la génération de l'îlot d'assemblage est terminée.

Ainsi, un îlot d'assemblage est généré par la dynamique que composent les requêtes, acceptations, refus, rejets, suicides et créations d'agents, et qui s'arrête lorsque tous les agents sont satisfaits, condition vérifiée indirectement par la condition de satisfaction technologique de l'agent-flot.

9.2.2 Parcours de l'espace physique

Le parcours de l'espace physique est géré uniquement par l'agent-flot: une fois qu'un îlot d'assemblage a été généré (donc que l'agent-flot est satisfait), il applique le recuit simulé. Sur la base de cette évaluation, il accepte ou non l'îlot généré comme le nouvel îlot courant. Puis, sur la base de l'îlot d'assemblage courant (qu'il ait changé ou non), il réactive les agents-postes, ce qui aura pour conséquence le rejet de certains agents-tâches: le SMA va alors tendre vers une nouvelle solution. La trajectoire du parcours de l'espace physique est ainsi constitué des îlots courants successifs de l'agent-flot.

L'optimum recherché est l'îlot d'assemblage dont le coût technologique est le moins cher. Le coût technologique d'un équipement dépend de sa nature, du type et du nombre de ses fonctions élémentaires constitutives. Le coût technologique d'un équipement est de la forme $c_e = b_e * q_e$ avec:

- b_e , le coût de base de l'équipement. Il dépend du type de l'équipement, du type et du nombre de ses fonctions élémentaires constitutives. Il représente le coût de fabrication de l'équipement.
- q_e , le facteur d'intégration de l'équipement. Il représente le coût additionnel engendré par l'intégration de l'équipement dans l'îlot d'assemblage. Il dépend de la nature de l'équipement.

Ainsi la fonction à minimiser est la suivante:

$$f_c(i) = \sum_{j=1}^{n_c} \left(\sum_{k=1}^{n_p^j} \left(\sum_{m=1}^{n_e^k} c_e^m \right) + c_{et}^j \right) + c_{et}^i \quad (9.1)$$

- i , l'îlot d'assemblage
- n_c , le nombre de cellules d'assemblage dans l'îlot d'assemblage,
- n_p^j , le nombre de postes d'assemblage dans la j -ième cellule d'assemblage,
- n_e^k , le nombre d'équipements constitutifs du k -ième poste d'assemblage,
- c_e^m , le coût technologique du m -ième équipement,
- c_{et}^j , le coût technologique du système de transfert de la j -ième cellule d'assemblage,
- c_{et}^i , le coût technologique du convoyeur libre de l'îlot d'assemblage

Chaque fois que l'agent-flot applique le recuit simulé, la probabilité d'acceptation du nouvel îlot d'assemblage est:

$$a(T^{a'}) = \begin{cases} 1 & \text{si } f_c(i_{new}^{a'}) \leq f_c(i_{crt}^{a'}) \\ \frac{f_c(i_{crt}^{a'}) T_{init}^{a'}}{f_c(i_{new}^{a'}) T_{crt}^{a'}} & \text{si } f_c(i_{new}^{a'}) > f_c(i_{crt}^{a'}) \end{cases}$$

avec:

- $i_{crt}^{a'}$, l'îlot d'assemblage courant;
- $i_{new}^{a'}$, l'îlot d'assemblage qui vient d'être déterminé;
- $T^{a'}$, la valeur de la tolérance au changement (température) de l'agent-flot;
- $T_{init}^{a'}$, la valeur initiale de la tolérance au changement de l'agent-flot;

Si l'îlot d'assemblage est accepté, il devient alors le nouvel îlot courant. En cas d'acceptation de $i_{new}^{a'}$ dans le cas où $f_c(i_{new}^{a'}) > f_c(i_{crt}^{a'})$, l'agent-flot diminue sa tolérance au changement.

Le parcours de l'espace physique s'arrête lorsqu'au moment de l'application du recuit simulé, la tolérance au changement $T^{a'}$ de l'agent-flot est nulle.

Ainsi, le critère de terminaison (selon la définition donnée au point 2.6) indiquant qu'un parcours de l'espace physique est achevé, est que tous les agents sont satisfaits et que la tolérance au changement de l'agent-flot est nulle.

En fait, le parcours décrit ci-dessus, ne constitue pas un parcours dans l'espace physique complet, mais dans un sous-espace de l'espace physique, déterminé par le processus opératoire p_{op} . Afin d'explorer un nouveau sous-espace, l'agent-flot a la possibilité alors de demander un nouveau processus d'assemblage à l'agent-processus. Il transforme ce nouveau processus d'assemblage en un processus opératoire, et il met à jour la population des agents-tâches avec pour conséquences la création de nouveaux agents-tâches et la disparition de ceux dont la condition d'existence n'est plus remplie: le SMA tend alors vers une solution et le parcours se poursuit selon la dynamique décrite ci-dessus.

Ainsi, un nouveau parcours de l'espace physique est possible, sur demande de l'utilisateur:

- soit en réinitialisant les paramètres du recuit simulé de l'agent-flot et en réactivant les agents-postes. On obtient ainsi une nouvelle trajectoire ne donnant pas forcément les mêmes résultats.
- soit en changeant de processus opératoire par l'envoi à l'agent-flot d'un message d'activation, avec pour conséquence un parcours dans un nouveau sous-espace de l'espace de physique, déterminé par le nouveau processus opératoire.

A tout moment du parcours de l'espace de recherche, l'agent-flot peut fournir à l'utilisateur l'ilot d'assemblage courant i_{crt}^a ou optimal i_{opt}^a du moment, sur simple demande de sa part.

9.3 Conclusion

Les objectifs du module de conception sont motivés par le parcours de l'espace physique de la conception des systèmes d'assemblage, constitué des flots d'assemblage admissibles. L'optimum recherché est l'ilot d'assemblage dont le coût technologique est le plus faible.

Le parcours de l'espace physique est obtenu par la dynamique des agents constituant le module de conception. L'agent-flot est chargé du contrôle de la trajectoire du parcours en appliquant le recuit simulé et en gérant les activations des autres agents. Le reste des agents (agents-tâches, agents-postes, agents-cellules) sont chargés de la spécification d'un ilot d'assemblage (équipements constitutifs et structure).

La génération d'un ilot d'assemblage, constituant un point de la trajectoire du parcours, est basée sur un processus opératoire regroupant les tâches constitutives (géométriques, physiques et complémentaires) et positionnelles (chargements, déchargements, réorientations) que doit réaliser l'ilot. Le processus d'assemblage, transformé en processus opératoire par l'agent-flot, est fourni par le module de sélection.

Un ilot d'assemblage est obtenu par les interactions entre les agents-tâches, les agents-postes, les agents-cellules et l'agent-flot: chaque agent cherche à remplir ses conditions de satisfaction et à maintenir ses conditions d'existence, en créant la partie d'ilot dont il est responsable (ressources génériques, poste, cellule) et en l'intégrant, en tant qu'élément, dans une structure adéquate (poste, cellule, flot). Les échanges de requêtes, acceptations, refus, et rejets, le suicide et la création dynamiques d'agents, s'arrêtent lorsque tous les agents sont satisfaits. L'ilot d'assemblage est généré.

Le parcours de l'espace de recherche s'arrête lorsque tous les agents du système sont satisfaits et que la tolérance au changement de l'agent-flot est nulle. Il est alors possible d'obtenir un nouveau parcours, en réactivant l'agent-flot ou en changeant de processus opératoire.

Les résultats du module de conception sont des spécifications d'ilots d'assemblage (optimaux ou non) qui décrivent les équipements constitutifs, leur organisation en postes et cellules, les temps de cycle et les coûts.

Chapitre 10

Exemple et Évaluation

10.1 Introduction

Les chapitres précédents présentaient les différentes modélisations utilisées dans AMACOIA, ainsi que la manière dont elles sont implantées et manipulées par le système.

Le but du présent chapitre est d'illustrer le point de vue de l'utilisateur et de discuter de la validité des modèles et des méthodes employées, au travers de l'exemple de la valve de vélo. Chaque module du système est analysé du point de vue de l'utilisateur, des résultats obtenus et des performances de l'approche.

Le système AMACOIA a été développé comme un outil à l'usage principal des concepteurs de systèmes d'assemblage; il doit, par conséquent, connaître la technologie et la terminologie du domaine de l'assemblage. En tant que tel, le système est alors utilisé soit comme support à la créativité de l'ingénieur afin d'explorer les différentes solutions intéressantes, soit pour obtenir rapidement une première spécification d'un flot d'assemblage.

Dans ce but, AMACOIA a été conçu pour s'adapter aux exigences et à l'expertise de l'utilisateur. Ainsi, le principe de base du système est la répartition des tâches entre l'expert humain et l'ordinateur.

Ce principe du partage des tâches est basé sur le fait suivant:

- les capacités d'analyse d'un expert humain sont supérieures (en qualité et en temps) à celles de l'ordinateur, alors que les capacités de calcul de l'ordinateur sont supérieures à celles de l'expert humain.

Dès lors les capacités d'analyse de l'expert humain sont utilisées pour restreindre, contraindre et définir l'espace de recherche sur lequel il souhaite travailler, alors que l'ordinateur est utilisé pour manipuler et explorer cet espace, sous la direction de l'utilisateur.

L'expert humain peut donc configurer le système, selon les capacités de raisonnement d'AMACOIA, la difficulté et les contraintes du problème à résoudre et le degré de précision souhaité pour les résultats. L'utilisateur fixant ainsi la répartition des tâches entre lui et l'ordinateur, le système s'adapte alors aux conditions imposés par l'humain.

10.2 Module d'initialisation

Dans le cadre du processus de conception des systèmes d'assemblage¹, le module d'initialisation du système AMACOIA couvre les étapes d'analyse du problème et la caractérisation des processus d'assemblage:

- la phase d'analyse du problème est laissée à l'expert. Au cours de cette phase, l'utilisateur fixe l'espace de recherche qu'il souhaite explorer;
- la caractérisation des processus d'assemblage (détermination du graphe ET/OU et découpage fonctionnel) est effectuée par le système.

Les entrées du module d'initialisation sont:

- une description du produit à assembler, représentée par l'arbre des références opératoires;
- un graphe de précédence décrivant les contraintes de précédence entre les réalisations des caractéristiques d'assemblage du produit. Ce graphe regroupe les différents processus d'assemblage du produit que souhaite évaluer l'utilisateur.

Ces entrées sont déterminées au cours de la phase d'analyse du produit.

Le résultat fourni par le module d'initialisation est le graphe ET/OU regroupant tous les processus d'assemblage admissibles retenus par l'utilisateur. Ce graphe ET/OU contient tous les sous-assemblages impliqués par les différents processus, ainsi que toutes les tâches d'assemblage qui les composent, découpées en opérations et en fonctions élémentaires.

10.2.1 Analyse du problème

Comme décrit au paragraphe 5.6, l'analyse du problème se base sur le cahier des charges et les différents dessins du produit à assembler, avec pour résultat, l'identification des contraintes et la modélisation du produit.

Le type des produits pouvant être traités par AMACOIA, dans son état actuel, sont les produits microtechniques que caractérisent leur taille relativement réduite (entre 1 et 300 mm) et leur cadence de production (temps de cycle inférieur à 10 sec). AMACOIA ne traite pour l'instant que les cas monoproduits. De plus, la modélisation géométrique des pièces ne permet de traiter que les pièces rigides dont la géométrie de base est prismatique.

Au cours de l'analyse, l'expert humain, sur la base du cahier des charges et des dessins du produit, est chargé d'identifier et de décrire au module d'initialisation les éléments suivants:

- le temps de cycle contrat;
- les éléments géométriques des pièces (faces élémentaires, dimensions) pertinents à l'assemblage: c'est-à-dire non seulement les éléments qui lui permettront de décrire les caractéristiques d'assemblage du produit, mais aussi ceux qui peuvent jouer un rôle au cours du processus d'assemblage (par exemple, les faces de positionnement et de saisie possibles);

¹ cf. section 5.5

- les caractéristiques géométriques d'assemblage qu'il souhaite voir réalisées;
- les caractéristiques physiques associées à chacune des caractéristiques géométriques;
- les caractéristiques complémentaires, incluant aussi bien les caractéristiques annexes du produit (peinture, marquage, usinage partiel, ...) que les caractéristiques informationnelles (test de qualité, réglage des mécanismes du produit, ...);
- les récepteurs et les directions principales de travail pour chacune des caractéristiques;
- les contraintes de précédence dans la réalisation des caractéristiques, contraintes exprimant aussi bien des solutions aux contraintes-produit (accessibilité, stabilité, et géométrie des constituants) que les critères stratégiques d'assemblage.

L'ensemble de ces éléments décrit et détermine l'espace fonctionnel des processus d'assemblage admissibles, sous-espace de l'espace complet des processus d'assemblage possibles, qui constituera l'entrée du système AMACIOA.

Le choix de se placer dès le départ dans ce sous-espace, plutôt que de traiter l'espace fonctionnel complet, est motivé par le fait que les temps de calcul sont ainsi réduits et surtout que, même si l'on traitait l'espace complet, la pratique a montré que seuls quelques processus d'assemblage sont retenus pour déterminer des flots d'assemblage. Il convient donc de se placer tout de suite dans le sous-espace le plus prometteur, tout en permettant à l'expert de le définir ou de le modifier selon son expérience.

L'identification des éléments permettant de décrire le sous-espace retenu, est structurée par la méthode de description imposée par le module d'initialisation. Cette méthode suit une approche constructive (bottom - up) allant du local au global, et permet à chaque étape de ne se poser que les questions pertinentes du moment. Elle se déroule en trois étapes:

1. la description des pièces. Chaque pièce est décrite en commençant par ses cotes, puis les faces élémentaires. Durant cette étape et pour chacune des pièces, l'utilisateur ne retient que les faces qu'il considère comme importantes pour le processus d'assemblage du produit. Ceci implique qu'il n'est pas obligé de décrire toutes les faces élémentaires, ni de les décrire précisément, puisqu'il peut décrire une abstraction de celles-ci, dont le comportement, du point de vue de l'assemblage, est identique à celui de la face élémentaire réelle.
2. la description des caractéristiques. Durant cette étape, l'expert humain choisit, localement à chaque caractéristique, le récepteur et la direction principale de travail. Cette étape termine la description de l'arbre des références opératoires.
3. la description des contraintes de précédence. Durant cette étape, l'utilisateur décrit le graphe de précédence.

10.2.2 Caractérisation des processus d'assemblage

Dans le module d'initialisation, cette étape est une reformulation des processus d'assemblage contenus dans la description faite par l'utilisateur. De ce fait, tous les processus d'assemblage décrits par l'utilisateur sont considérés par le système comme des processus d'assemblage admissibles, et la vérification de la validité de ces processus est laissée à l'utilisateur.

Le système détermine alors le graphe ET/OU représentant le sous-espace fonctionnel des processus d'assemblage admissibles sélectionné par l'utilisateur.

Le choix de représenter l'ensemble des processus d'assemblage admissibles par un graphe ET/OU, plutôt que par une autre représentation, a été motivé par le fait qu'il constitue une représentation compacte et explicite des processus d'assemblage retenus. De plus, son interprétation par l'utilisateur est immédiate puisqu'il est constitué de toutes les tâches et de tous les sous-assemblages impliqués par les processus d'assemblage, et ne nécessite pas d'effort d'abstraction pour se représenter ces éléments (comme pour un graphe de précedence, un graphe d'état ou un arbre d'assemblage qui ne contient pas de représentation explicite des sous-assemblages, par exemple).

Le graphe ET/OU étant déterminé, le système découpe alors fonctionnellement les tâches des processus d'assemblage, des tâches d'assemblage aux fonctions élémentaires.

10.2.3 Exemple

Le premier exemple porte sur la description de la soupape de la valve de vélo, dont le dessin est donné à la figure 10.1.

Sur la base du dessin, l'utilisateur commence par définir un repère ortho-normé, attaché à l'une des faces de la pièce. Ce repère est choisi de manière à pouvoir facilement décrire les cotes et correspond donc généralement aux axes de cotations de la pièce sur son dessin de détail.

Pour chacun des axes du repère de la pièce, il décrit alors les cotes fonctionnelles², les cotes auxiliaires³ étant calculées automatiquement par le système, avec leur tolérances, afin de faciliter par la suite la description des faces élémentaires.

Une cote est décrite par les deux points de l'axe du repère de la pièce qu'elle met en relation, sa valeur nominale, son écart inférieur et son écart supérieur. Un exemple de description d'une cote est donné dans la figure 10.2 en bas à droite: c'est la description du diamètre de $2.2 \text{ mm} \pm 0.05$ selon l'axe X du repère et centré sur son origine O_x . En tant que diamètre, cette cote est dupliquée automatiquement par le système sur l'axe Y.

Les cotes sont les principales informations géométriques utilisées dans AMA-COIA. Elles permettent:

- de décrire les dimensions des faces élémentaires des pièces;

²Ce sont les cotes qui déterminent les parties essentielles pour la fonction d'une pièce ou d'un sous-assemblage, de même que les cotes qui sont utilisées pour la fabrication et la vérification des pièces ou des sous-assemblage. Les cotes fonctionnelles doivent figurer sur les dessins de détails [71].

³Ce sont les cotes qui servent de complément aux cotes fonctionnelles pour éviter les calculs. Les cotes auxiliaires, si elles sont représentées sur les dessins, sont entre parenthèses.

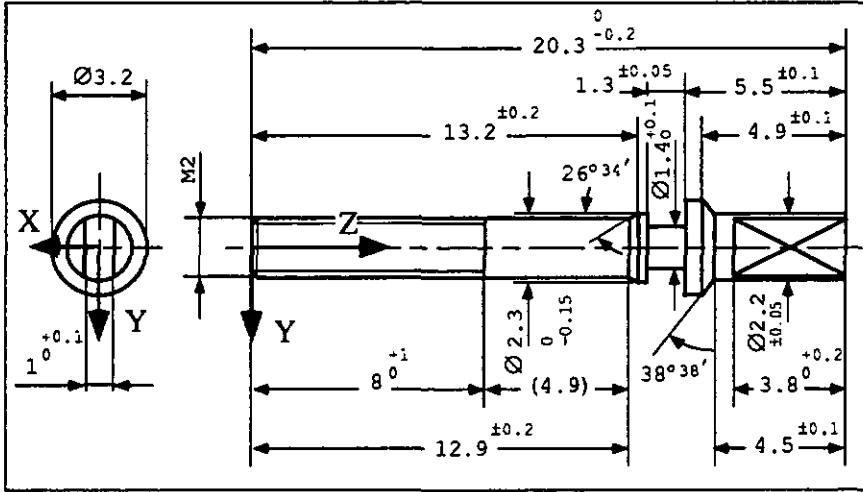


Figure 10.1: Dessin de détail de la soupape, avec le repère de la pièce

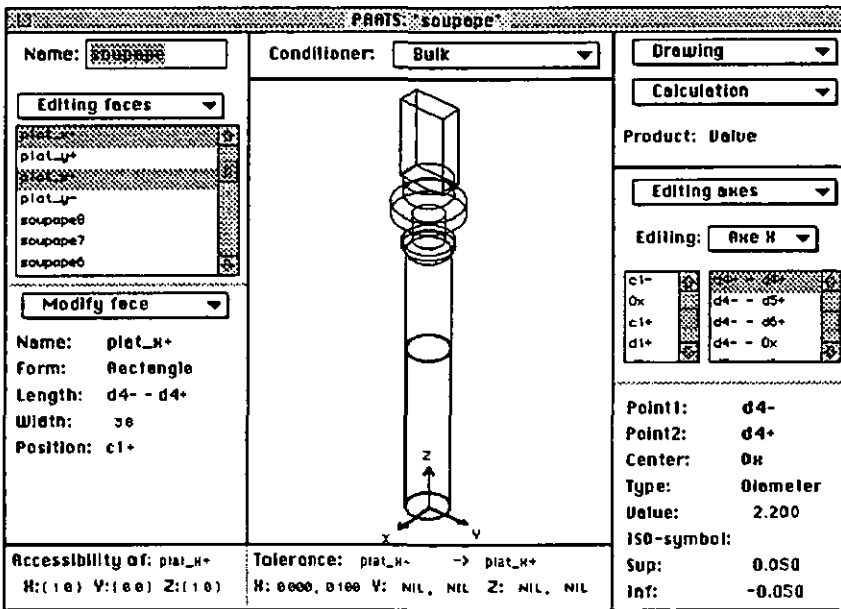


Figure 10.2: Fenêtre d'édition du composant élémentaire *soupape*: les éléments de $D^{soupape}$ sont donnés en bas à droite; les faces élémentaires $\in F^{soupape}$ sont donnés dans la partie gauche de la fenêtre; la représentation centrale permet de visualiser les faces élémentaires et les éléments de $G^{soupape}$

- de décrire les positions relatives des faces élémentaires par rapport au repère de la pièce;
- de décrire les positions relatives des pièces entre elles dans les caractéristiques géométriques;
- d'estimer les tolérances de positionnement des pièces les unes par rapport aux autres au sein d'un sous-assemblage, ou dans un posage. Remarque: il n'est pas tenu compte des tolérances angulaires dans ces estimations.

Une fois les cotes entrées, l'utilisateur décrit les différentes faces élémentaires de la pièce avec leur forme (rectangle, cylindre, cône, filetage, ...), leurs dimensions et leur position et orientation par rapport au repère de la pièce. A chaque face élémentaire est également associé un repère orthonormé par rapport auquel pourront être décrites les caractéristiques d'assemblage du produit.

Un exemple de description d'une face élémentaire est donné dans la figure 10.2 en bas à gauche. C'est la description de la face rectangulaire dont la normale pointe selon l'axe $X+$ du repère de la pièce: sa dimension selon l'axe Y correspond à la duplication de la cote $d4-$ - $d4+$ sur cet axe, sa dimension selon l'axe Z correspond à la cote de $3.8_0^{+0.2}$ et sa projection sur l'axe X (c'est une face plane) passe par le point $c1+$ de cet axe, point défini par la cote $c1-$ - $c1+$ de $1.1_0^{+0.1}$ de cet axe et centrée sur Ox .

Les faces élémentaires décrites sont les faces possibles pour le positionnement et la saisie des pièces et les faces de contact. Leur description n'a pas besoin de suivre précisément la géométrie réelle, mais peut être approximée, tout en gardant la même fonctionnalité du point de vue de l'assemblage. Par exemple, le bout de la soupape constitué de deux plats joints par les sections du cylindre dans lequel ils ont été usinés, a été décrit comme un parallépipède rectangle (cf. figure 10.2).

Sur la base de cette description des pièces, le système est capable de calculer les tolérances de positionnement des faces entre elles et de calculer l'accessibilité aux faces élémentaires, selon les 6 demi-axes du repère de la pièce.

La figure 10.2 donne un exemple de résultats pour ces calculs dans sa partie inférieure gauche. Ces résultats seront utilisés ultérieurement dans l'évaluation des processus d'assemblage et la caractérisation des équipements.

Une fois les pièces entrées, l'utilisateur décrit au système les caractéristiques d'assemblage. La principale difficulté de cette description consiste, pour l'utilisateur, à déterminer les récepteurs. Une méthode possible qu'il peut appliquer pour obtenir les récepteurs est de procéder par désassemblage du produit, sur la base du dessin d'ensemble en respectant les règles suivantes:

- on essaie de maximiser le nombre de sous-assemblages, c'est-à-dire que l'on privilégie le désassemblage d'un sous-ensemble de pièces par rapport au désassemblage d'une seule pièce. Remarque: dans le cadre de ce désassemblage, on considère comme sous-assemblage, un ensemble de caractéristiques d'assemblage (géométriques, physiques et/ou complémentaires) et les composants élémentaires sur lesquels elles portent;
- on désassemble une pièce ou un sous-assemblage dès que possible;
- on tient compte, au cours du désassemblage, des composants d'attachement, des caractéristiques physiques et des caractéristiques complémentaires;

- chaque sous-assemblage est désassemblé selon les mêmes principes, jusqu'à l'obtention de toutes les pièces disjointes.

Sur la base de ces règles on obtient un graphe de désassemblage, où un récepteur est la dernière pièce enlevée d'un sous-assemblage et du produit. La figure 10.3 donne l'exemple d'un graphe de désassemblage pour la valve de vélo où les récepteurs des caractéristiques géométriques sont le corps et la soupape, les récepteurs des caractéristiques physiques sont l'écrou, le guide et le joint, et le récepteur de la caractéristique complémentaire est la soupape.

Cette méthode offre aussi, pour l'utilisateur, l'avantage de pouvoir identifier une partie des contraintes de précédence dans la réalisation des caractéristiques.

Les figures 10.5 à 10.7 donnent des exemples de description de caractéristiques d'assemblage⁴:

- figure 10.5: fenêtre d'édition de la caractéristique géométrique d'assemblage g_a^3 représentant l'assemblage du guide dans le corps: les données sur la face du récepteur f_r^3 sont présentées en bas à gauche; f_c^3 , la face du composant est donnée en bas à droite; la visualisation centrale est fonction de M^3 , la matrice de coordonnées homogènes; ddl^3 , la matrice des degrés de liberté est donnée au-dessus de la représentation; dir^3 est représentée par la flèche dans la visualisation;
- figure 10.6: fenêtre d'édition de la caractéristique physique ϕ^2 , le chassage du guide: les données sur la face du récepteur f_r^2 sont présentées en bas à gauche; G^2 , l'ensemble des caractéristiques géométriques d'assemblage bloquées par ϕ^2 est donné en haut à droite; $proc^2$, le procédé de solidification utilisé est donné en haut à gauche; dir^2 est représentée par la flèche dans la visualisation;
- figure 10.7: fenêtre d'édition de la caractéristique complémentaire c_c^1 , le sertissage de la soupape: les données sur la face du récepteur f_r^1 sont présentées en bas à gauche; sub^1 , le sous-assemblage minimal est donné en haut à droite; dir^1 est représentée par la flèche dans la visualisation;

Une fois les caractéristiques d'assemblage décrites, il est possible d'obtenir la représentation de l'arbre des références opératoires (cf. figure 10.4), puisqu'il est entièrement déterminé par les données des caractéristiques.

Les dernières données introduites par l'utilisateur sont les contraintes de précédence sur la réalisation des caractéristiques d'assemblage. Ces contraintes sont regroupées dans un graphe de précédence (cf. figure 10.8).

L'espace fonctionnel de la conception des systèmes d'assemblage que souhaite explorer l'utilisateur est alors entièrement déterminé:

- la taille et le contenu des processus d'assemblage sont déterminés par le nombre et la définition des caractéristiques d'assemblage
- la taille de l'espace fonctionnel des processus d'assemblage exploré est déterminé par les alternatives qu'offrent les contraintes de précédence.

AMACIOA génère alors le graphe ET/OU représentant l'espace fonctionnel décrit, selon la méthode donnée au point 7.3 et découpe fonctionnellement ses tâches d'assemblage constitutives (cf. point 7.4).

⁴ Pour la description des caractéristiques et la terminologie voir aussi le point 7.1.

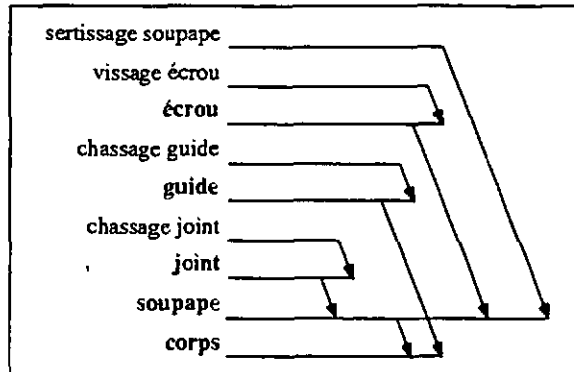


Figure 10.3: Graphe de désassemblage servant à la détermination des récepteurs des caractéristiques d'assemblage de la valve de vélo

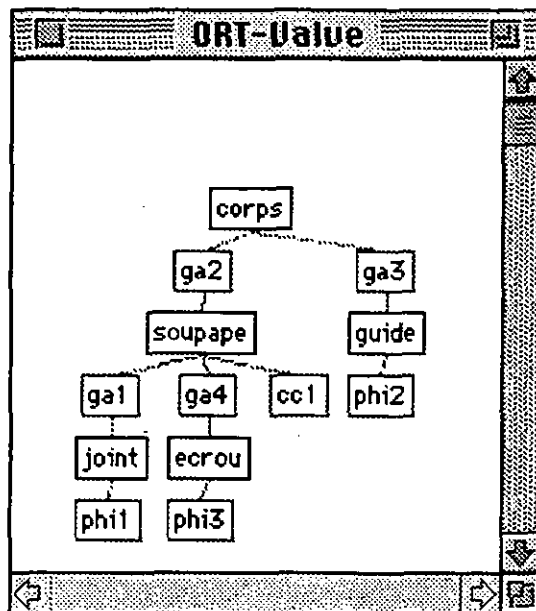


Figure 10.4: Fenêtre de visualisation de l'arbre des références opératoires de la valve

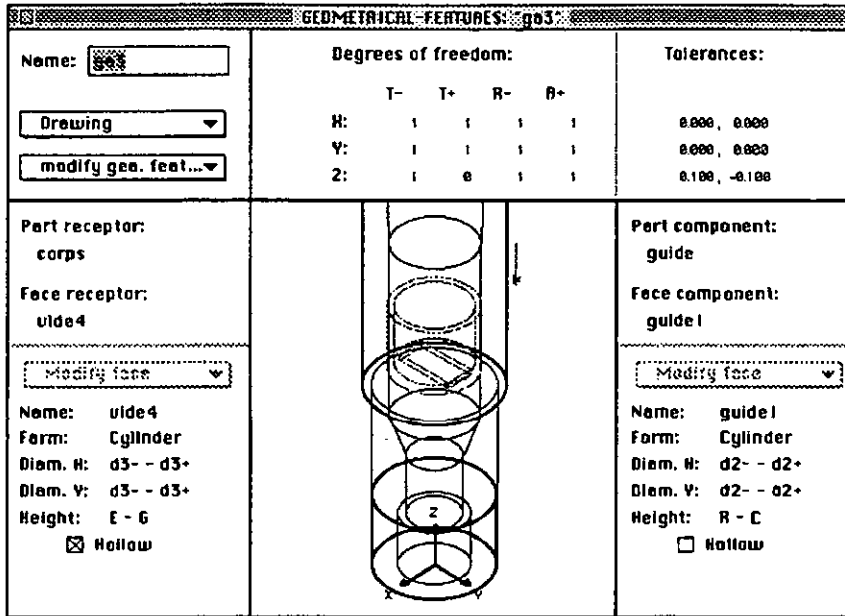


Figure 10.5: Fenêtre d'édition de la caractéristique géométrique d'assemblage g_a^3 entre le guide et le corps.

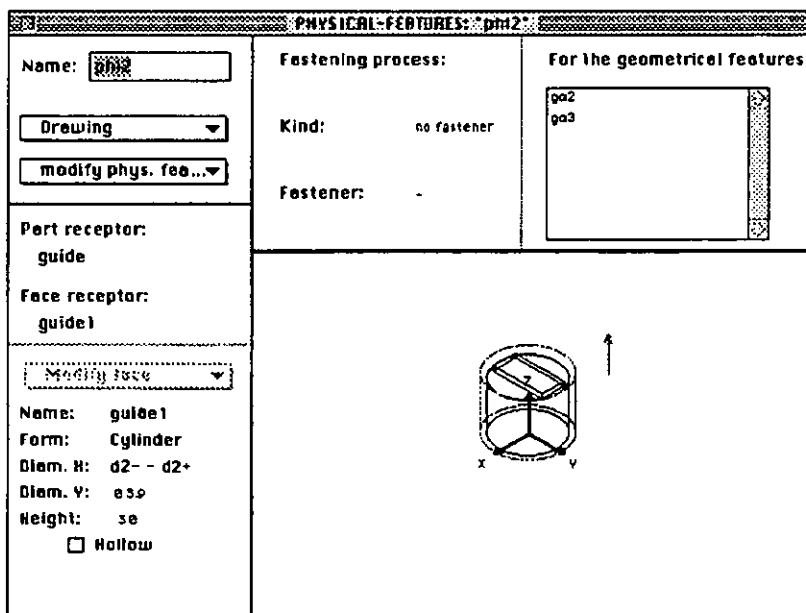


Figure 10.6: Fenêtre d'édition de la caractéristique physique ϕ^2 , le chassage du guide.

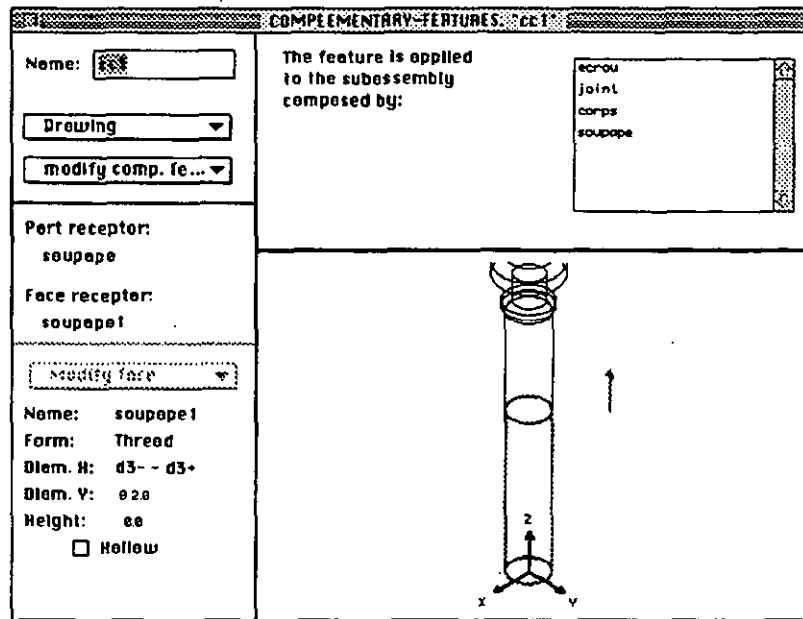


Figure 10.7: Fenêtre d'édition de la caractéristique complémentaire c_c^1 , le sertissage de la soupepe.

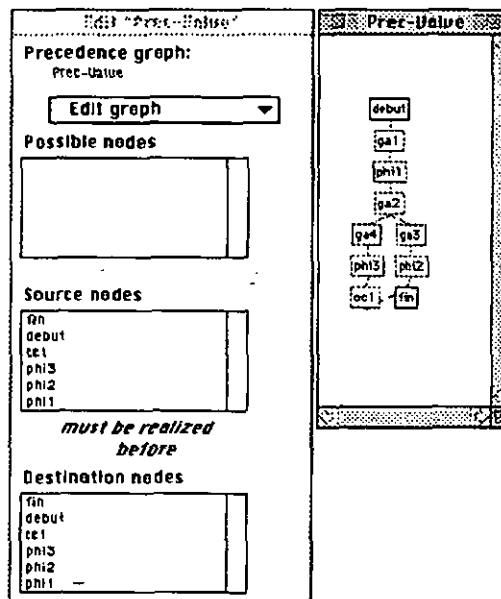


Figure 10.8: Fenêtres d'édition du graphe de précedence dans l'implantation du système AMACIOA: la fenêtre de gauche permet la description des contraintes de précedence et le graphe est tracé dans la fenêtre de droite

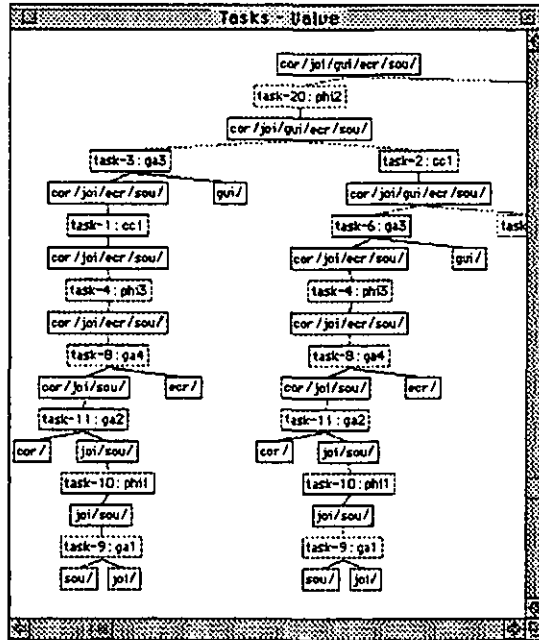


Figure 10.9: Fenêtres de visualisation du graphe ET/OU dans l'implantation du système AMACOIA. Seuls deux processus sont représentés.

10.2.4 Evaluation

Pour le module d'initialisation, l'approche consistant à exploiter au mieux les capacités d'analyse de l'expert humain offre les avantages suivants:

- la réduction de l'espace de recherche, en se focalisant dès le départ sur le sous-espace le plus prometteur (du point de vue de l'expert);
- la garantie de la cohérence des informations fournies: le système ne travaille que sur des processus d'assemblage admissibles impliquant des tâches et des sous-assemblages réalisables;
- la simplification des données introduites qui constituent alors une abstraction, du point de vue de l'assemblage, du produit réel.

Ceci a pour conséquence une simplification du moteur du module d'initialisation avec un gain sur les temps de calcul, par rapport à un système qui explorerait l'espace de recherche complet. Par contre, la nature des données introduites et l'intégration dans le modèle du produit de choix technologiques, impliquent que l'utilisateur potentiel d'AMACOIA doit être au fait du domaine de l'assemblage.

D'un autre côté, les capacités d'analyse de l'expert sont sous-utilisées lors du découpage fonctionnel des processus d'assemblage. En effet, le fait qu'il soit effectué uniquement par le système (dans son état actuel) ne semble pas satisfaisant. Il devrait pouvoir être fait interactivement par l'utilisateur, afin de pouvoir s'adapter aux cas non triviaux et d'améliorer les performances des modules de sélection et de conception.

L'introduction des données sous la forme d'un arbre des références opératoires et d'un graphe de précedence offre les avantages d'être simple et de correspondre à une démarche naturelle au cours de la phase d'analyse du problème. Par contre les données géométriques qu'ils regroupent ne sont pas pleinement utilisées par le système, ou incomplètes:

- les tolérances ne sont utilisées qu'à titre indicatif pour sélectionner les candidats potentiels parmi les faces de positionnement et de saisie, alors qu'elles peuvent jouer un rôle non négligeable dans le choix des équipements;
- certains éléments de la géométrie des pièces qui ont une importance en assemblage (comme les chanfreins, les états de surface, la rigidité des pièces), ne sont pas décrits (ou difficilement) et ne sont pas pris en compte dans le processus de conception.

Ceci a pour conséquence un appauvrissement de la qualité des solutions proposées, mais la prise en compte de ces éléments ne garantirait pas un gain pour le rapport entre qualité des solutions et performance du système.

L'effort principal dans l'implantation du module d'initialisation, a porté sur l'interface entre l'utilisateur et le système. Le but de cette interface est de permettre non seulement la saisie simple des données, mais aussi leur visualisation afin d'en vérifier la cohérence et leur manipulation afin d'explorer des hypothèses au cours de la phase d'analyse. Dans son implantation actuelle, l'interface ne remplit ces rôles que partiellement:

- la méthode de description de la géométrie est un peu fastidieuse (la saisie des données de la valve représente un peu moins d'une demi-journée de travail): pour la simplifier, on pourrait envisager l'utilisation d'un système de DAO couplé à AMACIOA, bien que dans la pratique les concepteurs de systèmes d'assemblage n'ont encore que le papier comme source principale d'information;
- la vérification de la cohérence des informations fournies par l'utilisateur (faisabilité géométrique des tâches, cohérence des données géométriques des caractéristiques, etc.) est laissée entièrement à l'utilisateur, alors qu'une partie de celle-ci pourrait être faite automatiquement par le système, sans trop pénaliser les temps de calcul: par exemple, la déduction de certaines contraintes de précedence, l'évaluation de la faisabilité des tâches, le choix des directions de travail, etc.

10.3 Module de sélection

Le module de sélection a pour but d'explorer l'espace des processus d'assemblage admissibles à la recherche du processus d'assemblage dont les tâches sont, en moyenne, les plus simples à réaliser (du point de vue fonctionnel).

L'entrée du module de sélection est le graphe ET/OU déterminé par le module d'initialisation et qui constitue l'espace à parcourir.

Les principaux résultats fournis par le module de sélection sont:

- soit les différents processus d'assemblage formant la trajectoire de l'exploration de l'espace de recherche;

- soit directement le processus d'assemblage optimal résultant d'un parcours de l'espace de recherche.

Ceci signifie que l'utilisateur peut soit suivre pas à pas l'exploration de l'espace de recherche, soit obtenir directement un processus d'assemblage optimal.

Dans le module de sélection, l'utilisateur n'a qu'un rôle d'observateur et ne peut pas influencer la recherche effectuée par le système.

10.3.1 Exemple

Une fois l'arbre des références opératoires et le graphe de précedence donnés, le système construit le graphe ET/OU qui contient tous les processus d'assemblage admissibles que souhaite explorer l'utilisateur.

Le système calcule, pour chaque tâche d'assemblage du graphe ET/OU, les valeurs de stabilité, d'accessibilité, le facteur de parallélisation et le facteur d'utilisation des récepteurs. Il calcule également la valeur de la fonction coût $f_d(p)$ pour les processus d'assemblage⁵:

$$f_d(p) = \sum_{k=1}^n d_t^k \quad (10.1)$$

- p , le processus d'assemblage
- n , le nombre de tâches d'assemblage dans le processus d'assemblage p
- d_t^k , la difficulté d'exécution estimée de la tâche constitutive k .
 $d_t^k = \frac{1}{4}(\sigma^k + \alpha^k + \pi^k + \rho^k)$

Remarque: Ces calculs se font au niveau des tâches d'assemblage et ne portent que sur les caractéristiques d'assemblage et les sous-assemblages impliqués. Ils ne prennent pas en compte le découpage fonctionnel des tâches d'assemblage, en opérations et en fonctions élémentaires.

L'utilisateur peut alors analyser les résultats de ces évaluations au moyen de dialogues:

- figure 10.10: cette fenêtre regroupe les résultats de l'évaluation des processus d'assemblage. L'utilisateur a la possibilité de consulter les valeurs des paramètres de $f_d(p)$ pour chaque processus d'assemblage, et d'afficher les processus;
- figure 10.11: fenêtre de visualisation d'une tâche d'assemblage qui regroupe toutes les informations sur la tâche, y compris une représentation des différents sous-assemblage impliqués par la tâche et ses opérations constitutives.

⁵ Voir l'annexe C pour les manières de calculer les paramètres.

Assembly Processes					
Product:	Value				
Process	Stab.	Access.	Parallel.	Recept.	f(p)
Proc-04	1.375	3.437	10.000	3.693	37.687
Proc-07	1.375	3.437	10.000	3.271	37.175
Proc-08	1.375	3.437	10.000	3.411	36.044
Proc-09	1.375	3.437	10.000	3.069	36.281
Proc-10	1.375	3.437	10.000	3.411	36.044

Best stability					
Proc-04	1.375	3.437	10.000	2.750	35.719

Best accessibility					
Proc-04	1.375	3.437	10.000	2.750	35.719

Best parallelization					
Proc-04	1.375	3.437	10.000	2.750	35.719

Best receptor using					
Proc-04	1.375	3.437	10.000	2.750	35.719

Best f(p)					
Proc-04	1.375	3.437	10.000	2.750	35.719

Figure 10.10: Fenêtre d'édition des processus d'assemblage.

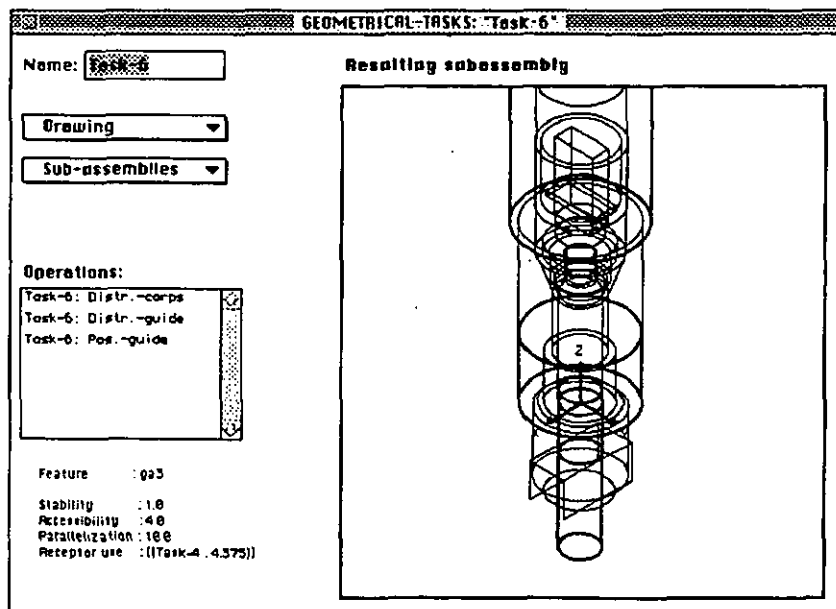


Figure 10.11: Fenêtre de visualisation d'une tâche d'assemblage.

Les tableaux 10.1 à 10.5 donnent des exemples de résultats obtenus pour la valve de vélo et auxquels peut accéder l'utilisateur au travers des dialogues:

- En ce qui concerne les processus (tableaux 10.1 et 10.3): en plus de la séquence des tâches formant le processus lui-même et sa valeur de $f_d(p)$, l'utilisateur dispose des valeurs moyennes de chacun des paramètres de $f_d(p)$ (stabilité, accessibilité, parallélisation, utilisation des récepteurs). Il peut en déduire alors les points faibles et les points forts du processus (du point de vue de la fonction coût).
- En ce qui concerne les tâches (tableaux 10.2, 10.4 et 10.5): pour chaque tâche, l'utilisateur dispose des valeurs de chacun des paramètres de $f_d(p)$ (stabilité, accessibilité, parallélisation, utilisation des récepteurs). Il peut en déduire alors quelle tâche constitue le point faible ou le point fort d'un processus et pourquoi (du point de vue de la fonction coût).

Sur la base de son analyse des résultats, l'utilisateur peut alors modifier et/ou affiner la description du sous-espace fonctionnel qu'il souhaite explorer en modifiant sa description (arbre des références opératoires et graphe de précedence), puis en contrôlant les conséquences du point de vue fonctionnel, jusqu'à un résultat satisfaisant.

la fonction coût $f_d(p)$

Le tableau 10.1 donne les valeurs moyennes de chacun des paramètres de la fonction coût et la valeur de cette fonction, pour chacun des processus d'assemblage contenus dans le graphe ET/OU de la figure 10.12.

La lecture de ces résultats montre que les facteurs de stabilité, d'accessibilité et de parallélisation ont la même valeur pour tous les processus d'assemblage, et que seul le facteur d'utilisation des récepteurs permet de différencier notablement les processus d'assemblage entre eux.

Si l'on regarde les valeurs des paramètres pour chaque tâche d'assemblage du graphe ET/OU (tableau 10.2), selon la caractéristique qu'elles réalisent, on constate que, pour une caractéristique donnée, les différentes tâches qui la réalisent (donc les alternatives possibles pour sa réalisation) ont des facteurs de stabilité, d'accessibilité et de parallélisation identiques, ce qui explique que les moyennes de ces paramètres pour les processus d'assemblage soient les mêmes.

Les raisons pour lesquelles les différentes tâches qui réalisent une caractéristique donnée ont des valeurs identiques, sont les suivantes:

- *Stabilité*: si le facteur de stabilité a la même valeur pour toutes les alternatives de réalisation d'une caractéristique donnée, cela est dû à la manière de calculer ce facteur. La stabilité estimée par ce facteur est la stabilité du récepteur dans le sous-assemblage récepteur concerné. Cette estimation ne tient pas compte des degrés de liberté du récepteur bloqués par le sous-assemblage lui-même, mais elle est calculée d'après la caractéristique géométrique d'assemblage dont la pièce est le composant. Cette caractéristique étant identique, quelles que soient les circonstances, la stabilité ne peut avoir que la même valeur.

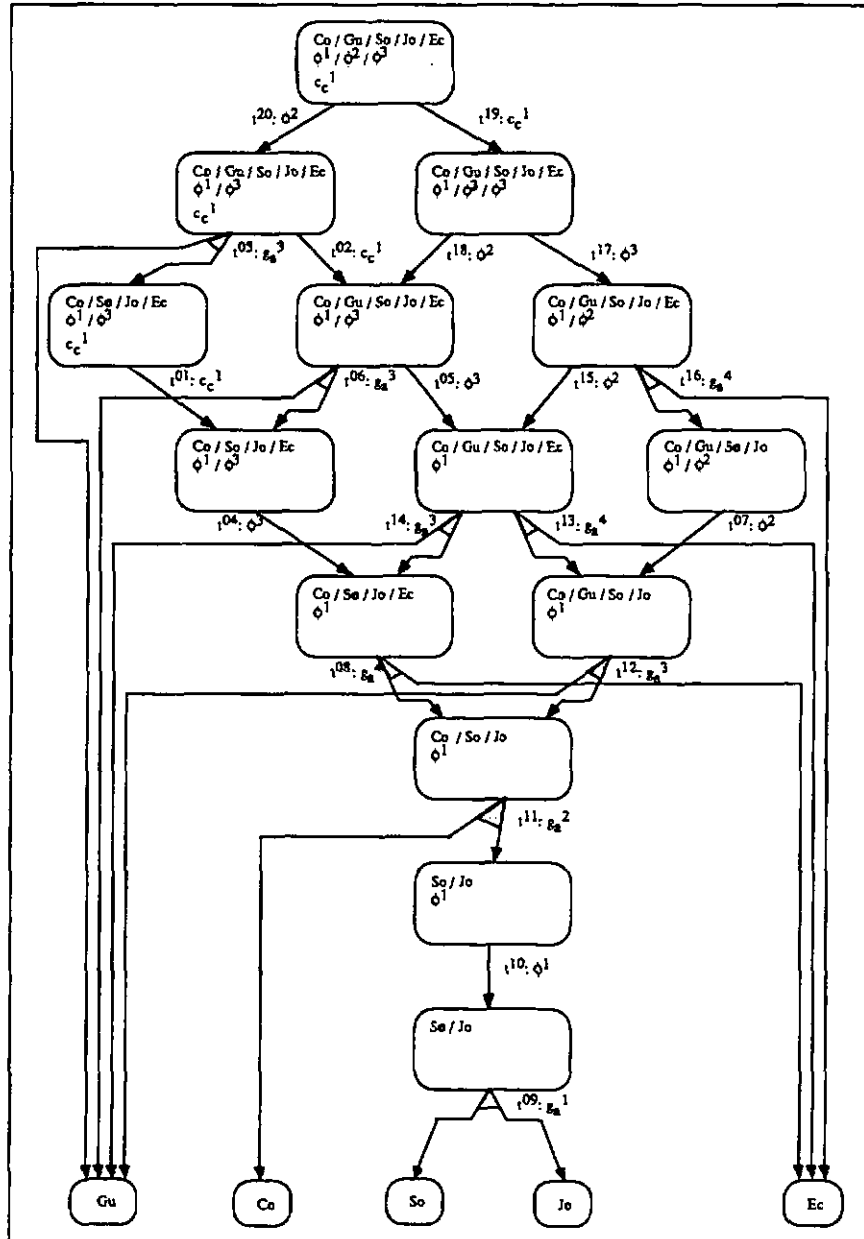


Figure 10.12: Graphe ET/OU représentant les processus d'assemblage de l'espace fonctionnel retenu pour la valve de vélo

Nom	Stab.	Acc.	Para.	Réc.	$f_d(p)$	Séquence
Proc-04	1.375	3.438	10.00	2.768	35.719	$t^{09} \rightarrow t^{10} \rightarrow t^{11} \rightarrow t^{12} \rightarrow t^{07} \rightarrow t^{16} \rightarrow t^{17} \rightarrow t^{19}$
Proc-09	1.375	3.438	10.00	3.089	36.281	$t^{09} \rightarrow t^{10} \rightarrow t^{11} \rightarrow t^{12} \rightarrow t^{13} \rightarrow t^{05} \rightarrow t^{02} \rightarrow t^{20}$
Proc-01	1.375	3.438	10.00	3.250	36.562	$t^{09} \rightarrow t^{10} \rightarrow t^{11} \rightarrow t^{08} \rightarrow t^{04} \rightarrow t^{01} \rightarrow t^{03} \rightarrow t^{20}$
Proc-08	1.375	3.438	10.00	3.410	36.844	$t^{09} \rightarrow t^{10} \rightarrow t^{11} \rightarrow t^{12} \rightarrow t^{13} \rightarrow t^{05} \rightarrow t^{18} \rightarrow t^{19}$
Proc-10	1.375	3.438	10.00	3.410	36.844	$t^{09} \rightarrow t^{10} \rightarrow t^{11} \rightarrow t^{12} \rightarrow t^{13} \rightarrow t^{15} \rightarrow t^{17} \rightarrow t^{19}$
Proc-02	1.375	3.438	10.00	3.571	37.125	$t^{09} \rightarrow t^{10} \rightarrow t^{11} \rightarrow t^{08} \rightarrow t^{04} \rightarrow t^{06} \rightarrow t^{18} \rightarrow t^{19}$
Proc-07	1.375	3.438	10.00	3.571	37.125	$t^{09} \rightarrow t^{10} \rightarrow t^{11} \rightarrow t^{08} \rightarrow t^{14} \rightarrow t^{15} \rightarrow t^{17} \rightarrow t^{19}$
Proc-03	1.375	3.438	10.00	3.893	37.688	$t^{09} \rightarrow t^{10} \rightarrow t^{11} \rightarrow t^{08} \rightarrow t^{04} \rightarrow t^{06} \rightarrow t^{02} \rightarrow t^{20}$
Proc-06	1.375	3.438	10.00	3.893	37.688	$t^{09} \rightarrow t^{10} \rightarrow t^{11} \rightarrow t^{08} \rightarrow t^{14} \rightarrow t^{05} \rightarrow t^{02} \rightarrow t^{20}$
Proc-05	1.375	3.438	10.00	4.214	38.250	$t^{09} \rightarrow t^{10} \rightarrow t^{11} \rightarrow t^{08} \rightarrow t^{14} \rightarrow t^{05} \rightarrow t^{18} \rightarrow t^{19}$

Tableau 10.1: Résultats de l'évaluation des processus d'assemblage issus du graphe de précédence de la figure 7.2 a).

Caractéristique	Tâche	Stab.	Acc.	Para.	Récepteur
ass. joint-soupape (g_a^1)	Task-09	1.000	4.000	10.00	-
chassage joint (ϕ^1)	Task-10	1.750	4.000	10.00	Task-09: 2.125
ass. soupape-corps (g_a^2)	Task-11	1.000	4.000	10.00	Task-10: 5.500
ass. guide-corps (g_a^3)	Task-03	1.000	4.000	10.00	Task-01: 4.375
	Task-06	1.000	4.000	10.00	Task-04: 4.375
	Task-12	1.000	4.000	10.00	Task-11: 1.000
	Task-14	1.000	4.000	10.00	Task-08: 4.375
chassage guide (ϕ^2)	Task-07	1.000	4.000	10.00	Task-12: 2.125
	Task-15	1.000	4.000	10.00	Task-13: 4.375
					Task-14: 2.125
	Task-18	1.000	4.000	10.00	Task-05: 4.375
	Task-20	1.000	4.000	10.00	Task-06: 2.125
				Task-02: 4.375	
				Task-03: 2.125	
ass. écrou-soupape (g_a^4)	Task-08	1.000	1.000	10.00	Task-11: 4.375
	Task-13	1.000	1.000	10.00	Task-12: 4.375
	Task-16	1.000	1.000	10.00	Task-07: 4.375
vissage écrou (ϕ^3)	Task-04	3.250	5.500	10.00	Task-08: 2.125
	Task-05	3.250	5.500	10.00	Task-13: 2.125
					Task-14: 4.375
	Task-17	3.250	5.500	10.00	Task-15: 4.375
					Task-16: 2.125
sertissage soupape (c_c^1)	Task-01	1.000	1.000	10.00	Task-04: 2.125
	Task-02	1.000	1.000	10.00	Task-05: 2.125
					Task-06: 4.375
	Task-19	1.000	1.000	10.00	Task-17: 2.125
				Task-18: 4.375	

Tableau 10.2: Tâches alternatives pour la réalisation des caractéristiques de la valve

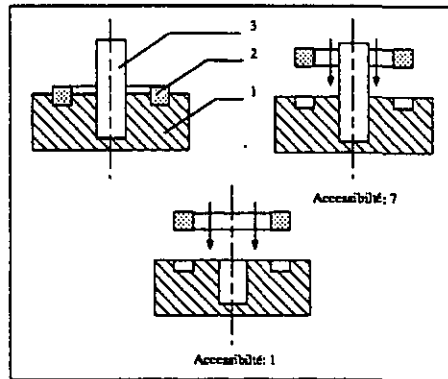


Figure 10.13: Exemples de cas où le facteur d'accessibilité permet de différencier deux tâches réalisant une caractéristique donnée (ici l'assemblage de l'anneau sur le corps)

- **Accessibilité:** en ce qui concerne le facteur d'accessibilité, le fait que sa valeur soit la même provient de la géométrie de la valve. L'assemblage de la valve peut être considéré comme un empilement de pièces dans le corps selon un ordre fixé par les contraintes de précédence (joint - soupape - guide, d'un côté; écrou, de l'autre). Ainsi, les circonstances géométriques de l'assemblage d'une pièce donnée (donc la réalisation d'une caractéristique géométrique donnée) ne peuvent pas varier et l'accessibilité au récepteur de la caractéristique ne peut pas varier d'une tâche à l'autre. La figure 10.13 donne un exemple de produit où la valeur de l'accessibilité peut varier d'une tâche à l'autre pour une caractéristique donnée.
- **Parallélisation:** en ce qui concerne le facteur de parallélisation, le fait que sa valeur soit la même provient de la structure de la valve. La valve n'ayant pas de sous-assemblages susceptibles d'être réalisés en parallèle avant d'être assemblés dans un montage final, le facteur de parallélisation de toutes les tâches ne peut être égal qu'à 10.

Le deuxième test effectué sur la valve permet de se rendre compte que la fonction coût seule n'est pas suffisante pour déterminer si un processus est admissible ou non. Les conditions de ce test sont les suivantes:

- pour des raisons de taille d'espace et de représentation, les caractéristiques physiques de la valve n'ont pas été prises en compte, avec pour conséquence que le calcul de la stabilité s'en trouve partiellement faussé, puisqu'il tient compte de la réalisation ou non des caractéristiques physiques dans les sous-assemblages concernés;
- il n'y a aucune contrainte de précédence dans la réalisation des caractéristiques. Ceci implique que les processus d'assemblage générés peuvent ne pas être admissibles, de même que certains processus peuvent contenir des branches pouvant se réaliser en parallèle.

La figure 10.14 donne le graphe de précédence et l'arbre des références opératoires retenus pour le second test. Le graphe ET/OU résultant est donné à la figure 10.15.

Les résultats de l'évaluation des processus d'assemblage, donnés dans les tables 10.3, 10.4 et 10.5, montrent les faits suivants:

- concernant les processus d'assemblage: les processus admissibles (Proc-39, Proc-41 et Proc-42) ne font pas partie des processus optimaux selon la fonction coût. Ceci signifie que la fonction coût $f_d(p)$ ne peut pas être utilisée pour la sélection des processus admissibles dans l'espace fonctionnel. Ceci est acceptable puisque l'hypothèse de base avance que l'espace fonctionnel exploré par le module de sélection ne devrait être composé que de processus d'assemblage admissibles.

Dans ce test, l'ensemble des processus générés constitue l'espace fonctionnel complet de l'exemple, avec seulement trois processus admissibles (soit 7 pour cent). Ceci montre qu'il est donc préférable, pour des raisons de temps de calcul, de se placer directement dans le sous-espace des processus admissibles, d'autant plus que l'on peut en augmenter la précision de description (comme dans le premier test, par exemple) avec un gain de précision sur les résultats ultérieurs.

- concernant la stabilité: elle n'a pas les mêmes valeurs pour les alternatives de réalisation d'une caractéristique donnée. Cela provient de la manière de calculer ce paramètre, et du fait que ce calcul n'est plus soumis aux contraintes de précédence dans ce test.
- concernant l'accessibilité: elle a toujours une valeur identique pour les mêmes raisons évoquées dans le premier test, valeur d'ailleurs identique dans les deux cas.

Remarque: dans le cas des processus d'assemblage dont certaines tâches s'exécutent en parallèle (reconnaisables à un facteur de parallélisation différent de 10), le facteur d'utilisation des récepteurs de la tâche aval est le facteur le plus défavorable. Par exemple pour la tâche Task-35 qui succède aux tâches Task-17 et Task-04 exécutées en parallèle dans le processus Proc-11 (voir tableau 10.3), le facteur d'utilisation des récepteurs vaut 5.50 (voir tableau 10.4), ce qui correspond à la tâche Task-17, alors que celui qui correspond à la tâche Task-04 vaut 1.0, ce qui est plus favorable.

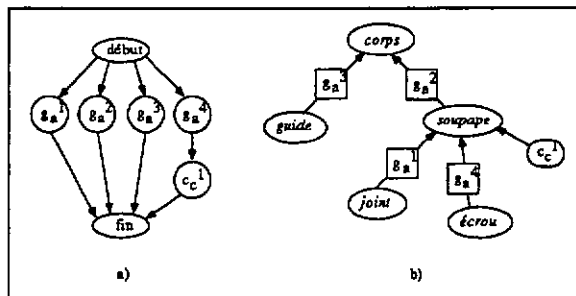


Figure 10.14: Graphe de précédence a) et arbre des références opératoires b) pour le second test sur la valve

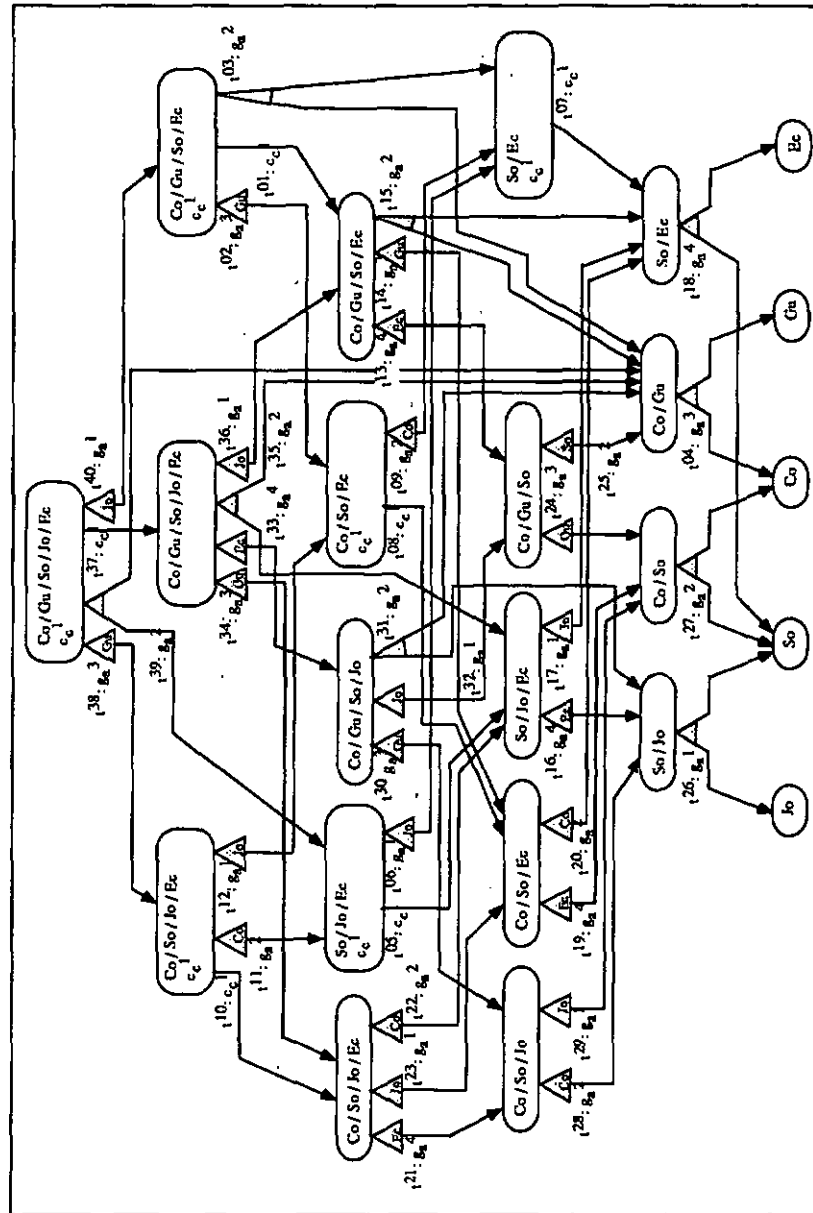


Figure 10.15: Graphe ET/OU regroupant les 42 processus d'assemblage du second test. Les arcs des hyperarcs ET pointant sur des composants élémentaires ne sont pas représentés, mais remplacés par les initiales du composant dans le triangle de l'hyperarc ET

Nom	Stab.	Acc.	Para.	Réc.	$f_d(p)$	Séquence
Proc-02	1.000	2.800	9.100	2.500	20.250	$(t^{18} \rightarrow t^{07} \rightarrow t^{06})/(t^{04}) \rightarrow t^{39}$
Proc-07	1.000	2.800	9.100	2.500	20.250	$(t^{18} \rightarrow t^{17} \rightarrow t^{05})/(t^{04}) \rightarrow t^{39}$
Proc-19	1.000	2.800	9.100	2.500	20.250	$(t^{26} \rightarrow t^{16} \rightarrow t^{05})/(t^{04}) \rightarrow t^{39}$
Proc-08	1.000	2.800	10.00	2.125	20.375	$t^{18} \rightarrow t^{07} \rightarrow t^{06} \rightarrow t^{11} \rightarrow t^{38}$
Proc-20	1.000	2.800	10.00	2.125	20.375	$t^{18} \rightarrow t^{17} \rightarrow t^{05} \rightarrow t^{11} \rightarrow t^{38}$
Proc-37	1.000	2.800	10.00	2.125	20.375	$t^{26} \rightarrow t^{16} \rightarrow t^{05} \rightarrow t^{11} \rightarrow t^{38}$
Proc-04	2.200	2.800	8.200	3.625	21.469	$(t^{18})/(t^{04}) \rightarrow t^{15} \rightarrow t^{36} \rightarrow t^{37}$
Proc-06	2.200	2.800	8.200	3.625	21.469	$(t^{18})/(t^{04}) \rightarrow t^{15} \rightarrow t^{01} \rightarrow t^{40}$
Proc-29	2.200	2.800	8.200	3.625	21.469	$(t^{26})/(t^{04}) \rightarrow t^{31} \rightarrow t^{33} \rightarrow t^{37}$
Proc-01	1.600	2.800	9.100	3.625	21.844	$(t^{18} \rightarrow t^{07})/(t^{04}) \rightarrow t^{03} \rightarrow t^{40}$
Proc-11	1.600	2.800	9.100	3.625	21.844	$(t^{18} \rightarrow t^{17})/(t^{04}) \rightarrow t^{35} \rightarrow t^{37}$
Proc-25	1.600	2.800	9.100	3.625	21.844	$(t^{26} \rightarrow t^{16})/(t^{04}) \rightarrow t^{35} \rightarrow t^{37}$
Proc-05	1.600	2.800	10.00	2.969	21.969	$t^{18} \rightarrow t^{07} \rightarrow t^{09} \rightarrow t^{02} \rightarrow t^{40}$
Proc-26	1.600	2.800	10.00	2.969	21.969	$t^{18} \rightarrow t^{17} \rightarrow t^{22} \rightarrow t^{34} \rightarrow t^{37}$
Proc-40	1.600	2.800	10.00	2.969	21.969	$t^{26} \rightarrow t^{16} \rightarrow t^{22} \rightarrow t^{34} \rightarrow t^{37}$
Proc-13	2.800	2.800	10.00	1.844	22.594	$t^{04} \rightarrow t^{25} \rightarrow t^{32} \rightarrow t^{33} \rightarrow t^{37}$
Proc-15	2.800	2.800	10.00	1.844	22.594	$t^{04} \rightarrow t^{25} \rightarrow t^{13} \rightarrow t^{36} \rightarrow t^{37}$
Proc-18	2.800	2.800	10.00	1.844	22.594	$t^{04} \rightarrow t^{25} \rightarrow t^{13} \rightarrow t^{01} \rightarrow t^{40}$
Proc-31	2.800	2.800	10.00	1.844	22.594	$t^{27} \rightarrow t^{24} \rightarrow t^{32} \rightarrow t^{33} \rightarrow t^{37}$
Proc-33	2.800	2.800	10.00	1.844	22.594	$t^{27} \rightarrow t^{24} \rightarrow t^{13} \rightarrow t^{36} \rightarrow t^{37}$
Proc-36	2.800	2.800	10.00	1.844	22.594	$t^{27} \rightarrow t^{24} \rightarrow t^{13} \rightarrow t^{01} \rightarrow t^{40}$
Proc-14	2.200	2.800	10.00	2.969	22.719	$t^{18} \rightarrow t^{20} \rightarrow t^{14} \rightarrow t^{36} \rightarrow t^{37}$
Proc-17	2.200	2.800	10.00	2.969	22.719	$t^{18} \rightarrow t^{20} \rightarrow t^{14} \rightarrow t^{01} \rightarrow t^{40}$
Proc-42	2.200	2.800	10.00	2.969	22.719	$t^{26} \rightarrow t^{28} \rightarrow t^{30} \rightarrow t^{33} \rightarrow t^{37}$
Proc-03	1.600	2.800	10.00	3.813	22.813	$t^{18} \rightarrow t^{07} \rightarrow t^{09} \rightarrow t^{12} \rightarrow t^{38}$
Proc-21	1.600	2.800	10.00	3.813	22.813	$t^{18} \rightarrow t^{17} \rightarrow t^{22} \rightarrow t^{10} \rightarrow t^{38}$
Proc-38	1.600	2.800	10.00	3.813	22.813	$t^{26} \rightarrow t^{16} \rightarrow t^{22} \rightarrow t^{10} \rightarrow t^{38}$
Proc-22	2.800	2.800	10.00	2.688	23.438	$t^{27} \rightarrow t^{29} \rightarrow t^{21} \rightarrow t^{10} \rightarrow t^{38}$
Proc-23	2.800	2.800	10.00	2.688	23.438	$t^{27} \rightarrow t^{19} \rightarrow t^{23} \rightarrow t^{10} \rightarrow t^{38}$
Proc-24	2.800	2.800	10.00	2.688	23.438	$t^{27} \rightarrow t^{19} \rightarrow t^{08} \rightarrow t^{12} \rightarrow t^{38}$
Proc-09	2.200	2.800	10.00	3.812	23.563	$t^{18} \rightarrow t^{20} \rightarrow t^{23} \rightarrow t^{10} \rightarrow t^{38}$
Proc-10	2.200	2.800	10.00	3.812	23.563	$t^{18} \rightarrow t^{20} \rightarrow t^{08} \rightarrow t^{12} \rightarrow t^{38}$
Proc-39	2.200	2.800	10.00	3.812	23.563	$t^{26} \rightarrow t^{28} \rightarrow t^{21} \rightarrow t^{10} \rightarrow t^{38}$
Proc-27	2.800	2.800	10.00	3.531	24.281	$t^{27} \rightarrow t^{29} \rightarrow t^{21} \rightarrow t^{34} \rightarrow t^{37}$
Proc-28	2.800	2.800	10.00	3.531	24.281	$t^{27} \rightarrow t^{19} \rightarrow t^{23} \rightarrow t^{34} \rightarrow t^{37}$
Proc-30	2.800	2.800	10.00	3.531	24.281	$t^{27} \rightarrow t^{29} \rightarrow t^{30} \rightarrow t^{33} \rightarrow t^{37}$
Proc-32	2.800	2.800	10.00	3.531	24.281	$t^{27} \rightarrow t^{19} \rightarrow t^{14} \rightarrow t^{36} \rightarrow t^{37}$
Proc-34	2.800	2.800	10.00	3.531	24.281	$t^{27} \rightarrow t^{19} \rightarrow t^{08} \rightarrow t^{02} \rightarrow t^{40}$
Proc-35	2.800	2.800	10.00	3.531	24.281	$t^{27} \rightarrow t^{19} \rightarrow t^{14} \rightarrow t^{01} \rightarrow t^{40}$
Proc-12	2.200	2.800	10.00	4.656	24.406	$t^{18} \rightarrow t^{20} \rightarrow t^{24} \rightarrow t^{33} \rightarrow t^{37}$
Proc-16	2.200	2.800	10.00	4.656	24.406	$t^{18} \rightarrow t^{20} \rightarrow t^{08} \rightarrow t^{02} \rightarrow t^{40}$
Proc-41	2.200	2.800	10.00	4.656	24.406	$t^{26} \rightarrow t^{28} \rightarrow t^{21} \rightarrow t^{34} \rightarrow t^{37}$

Tableau 10.3: Résultats de l'évaluation des processus d'assemblage issus du graphe de précedence de la figure 10.14 a)

Caractéristique	Tâche	Stab.	Acc.	Para.	Récepteur
ass. joint-soupape (g_a^1)	Task-06	1.000	4.000	10.00	Task-07: 1.000
	Task-12	4.000	4.000	10.00	Task-08: 1.000
					Task-09: 4.375
	Task-17	1.000	4.000	10.00	Task-18: 1.000
	Task-23	4.000	4.000	10.00	Task-19: 1.000
					Task-20: 4.375
	Task-26	1.000	4.000	10.00	-
	Task-29	4.000	4.000	10.00	Task-27: 4.375
	Task-32	4.000	4.000	10.00	Task-24: 4.375
					Task-25: 4.375
	Task-36	4.000	4.000	10.00	Task-13: 1.000
					Task-14: 4.375
					Task-15: 4.375
	Task-40	4.000	4.000	10.00	Task-01: 1.000
				Task-02: 4.375	
				Task-03: 4.375	
ass. soupape-corps (g_a^2)	Task-03	1.000	4.000	5.500	Task-04: 1.000
					Task-07: 5.500
	Task-09	1.000	4.000	10.00	Task-07: 5.500
	Task-11	1.000	4.000	10.00	Task-05: 5.500
					Task-06: 5.500
	Task-15	1.000	4.000	1.00	Task-04: 1.000
					Task-18: 5.500
	Task-20	1.000	4.000	10.00	Task-18: 5.500
	Task-22	1.000	4.000	10.00	Task-16: 5.500
					Task-17: 5.500
	Task-25	1.000	4.000	10.00	Task-04: 1.000
	Task-27	1.000	4.000	10.00	-
	Task-28	1.000	4.000	10.00	Task-26: 5.500
	Task-31	1.000	4.000	1.00	Task-04: 1.000
					Task-26: 5.500
Task-35	1.000	4.000	5.50	Task-04: 1.000	
				Task-16: 5.500	
				Task-17: 5.500	
Task-39	1.000	4.000	5.50	Task-04: 1.000	
				Task-05: 5.500	
				Task-06: 5.500	

Tableau 10.4: Tâches alternatives pour la réalisation des caractéristiques de la valve

Caractéristique	Tâche	Stab.	Acc.	Para.	Récepteur
ass. guide-corps (g_a^3)	Task-02	1.000	4.000	10.00	Task-08: 4.375 Task-09: 1.000
	Task-04	1.000	4.000	10.00	-
	Task-14	1.000	4.000	10.00	Task-19: 4.375 Task-20: 1.000
	Task-24	1.000	4.000	10.00	Task-27: 1.000
	Task-30	1.000	4.000	10.00	Task-28: 1.000 Task-29: 4.375
	Task-34	1.000	4.000	10.00	Task-21: 4.375 Task-22: 1.000 Task-23: 4.375
	Task-38	1.000	4.000	10.00	Task-10: 4.375 Task-11: 1.000 Task-12: 4.375
	Task-13	4.000	1.000	10.00	Task-24: 4.375 Task-25: 4.375
	Task-16	1.000	1.000	10.00	Task-26: 1.000
	Task-18	1.000	1.000	10.00	-
ass. écrou-soupape (g_a^4)	Task-19	4.000	1.000	10.00	Task-27: 4.375
	Task-21	4.000	1.000	10.00	Task-28: 4.375 Task-29: 1.000
	Task-33	4.000	1.000	10.00	Task-30: 4.375 Task-31: 4.375 Task-32: 1.000
	Task-01	4.000	1.000	10.00	Task-13: 1.000 Task-14: 4.375 Task-15: 4.375
	Task-05	1.000	1.000	10.00	Task-16: 1.000 Task-17: 1.000
	Task-07	1.000	1.000	10.00	Task-18: 1.000
	Task-08	4.000	1.000	10.00	Task-19: 1.000 Task-20: 4.375
sertissage soupape (c_c^1)	Task-10	4.000	1.000	10.00	Task-21: 1.000 Task-22: 4.375 Task-23: 1.000
	Task-37	4.000	1.000	10.00	Task-33: 1.000 Task-34: 4.375 Task-35: 4.375 Task-36: 1.000

Tableau 10.5: Tâches alternatives pour la réalisation des caractéristiques de la valve

Optimization and Design					
Process	Stab.	Access.	Paralel.	Recept.	f(p)
Proc-01	1.575	3.437	10.000	3.250	36.562
Proc-02	1.575	3.437	10.000	3.571	37.125
Proc-03	1.575	3.437	10.000	3.693	37.687
Proc-04	1.575	3.437	10.000	2.768	35.719
Proc-05	1.575	3.437	10.000	4.214	38.250
Proc-06	1.575	3.437	10.000	3.693	37.687
Proc-07	1.575	3.437	10.000	3.571	37.125

Product:

Draw proc.

Functional Optimization: **Assembly line design:**

Optimize Design

Solution reached Solution reached

Value of the current proc.: 37.125 Cost of the line:

Value of the optimal proc.: 35.719 Cycle time of the line:

Figure 10.16: Fenêtre de contrôle des systèmes multi-agents

Dynamique des agents

Toute l'analyse des évaluations et les modifications de la description de l'espace fonctionnel que souhaite explorer l'utilisateur, et qui ont été présentées dans les paragraphes précédents, peuvent se faire indépendamment du système multi-agents du module.

Dans une seconde phase l'utilisateur a alors recours au système multi-agents pour sélectionner le processus d'assemblage (optimal ou non) sur lequel il veut concevoir les systèmes d'assemblage: à chaque étape de l'exploration de l'espace par le système multi-agent, il peut choisir de fournir le processus d'assemblage courant (ou le processus optimal du moment) au module de conception.

La figure 10.16 montre la fenêtre de contrôle des deux systèmes multi-agents. La partie de gauche sert pour le module de sélection ("Functional Optimization") et la partie de droite ("Design") pour le module de conception.

La figure 10.17 montre l'évolution des populations d'agents-tâches satisfaits et bloqués au cours du temps, pour la sélection de 4 processus d'assemblage successifs.

L'échelle de temps n'est pas linéaire, mais proportionnelle au nombre de requêtes qu'a reçues l'agent-processus: les mesures de ces populations ont été faites par l'agent-processus, chaque fois qu'il recevait une requête de la part d'un agent-tâche.

On constate les faits suivants:

- le temps nécessaire pour sélectionner un processus d'assemblage n'est pas constant, ce qui provient du caractère aléatoire du système multi-agent: les échanges de messages entre agents sont asynchrones et leur activation se fait de manière aléatoire.

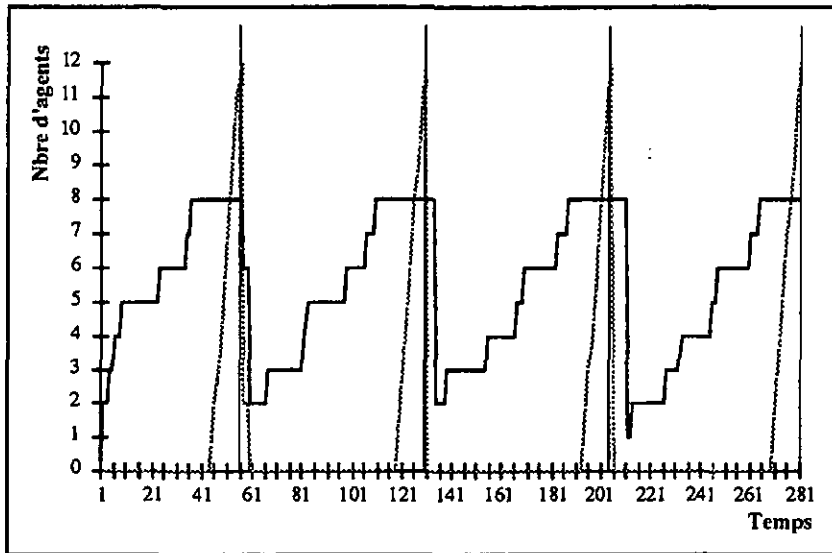


Figure 10.17: Dynamique des populations d'agents au cours du temps: en noir, nombre d'agents-tâches satisfaits; en gris, nombre d'agents-tâches bloqués

- la population des agents-tâches satisfaits croît par paliers plus ou moins longs au cours de la sélection d'un processus. La longueur de ces paliers dépend en fait du nombre de requêtes rejetées entre deux acceptations.
- la croissance de la population des agents-tâches satisfaits cesse au moment où il y a 8 agents satisfaits. Ce chiffre correspond en fait au nombre de tâches composant un processus d'assemblage pour la valve.
- la population des agents-tâches bloqués croît dès que le nombre d'agents-tâches satisfaits atteint son maximum (avec un petit décalage dû à la méthode de mesure), ce qui correspond au moment où l'agent-processus a un processus complet. Cette population croît jusqu'à 12, moment où tous les agents-tâches du système (au nombre total de 20) sont soit satisfaits, soit bloqués: l'agent-processus est satisfait et le processus est sélectionné.
- au moment où commence la sélection d'un nouveau processus, il peut s'écouler un instant avant que des agents-tâches ne soient rejetés et perdent leur satisfaction. Cette période correspond au nombre de requêtes qui ne représentent pas une alternative pour le processus courant. Par contre dès qu'une alternative est possible, la population d'agents satisfaits décroît rapidement, mais sans tomber à zéro. Ceci correspond aux rejets des agents-tâches en aval de l'alternative, avant que l'agent-processus ne recommence à construire un nouveau processus à partir des tâches restantes.

Du point de vue performance, la dynamique multi-agent semble ne pas apporter de gain de temps (ce qui n'est pas démontré dans ce travail). Par contre, c'est une méthode efficace et simple pour obtenir une réorganisation dynamique des solutions au cours de l'exploration de l'espace.

10.3.2 Evaluation

La fonction coût $f_d(p)$ montre quelques faiblesses dans la caractérisation des processus d'assemblage:

- les facteurs de stabilité, d'accessibilité et de parallélisation, sous leur forme actuelle, permettent davantage d'évaluer l'assemblabilité d'un produit, plutôt que de caractériser des processus d'assemblage. Il faudrait donc modifier leur calcul afin de mieux intégrer l'aspect processus, par exemple en se basant plutôt sur les sous-assemblages concernés que sur les caractéristiques d'assemblage du produit.
- elle ne porte que sur la difficulté de réalisation des tâches d'assemblage. En introduisant dans la fonction de nouveaux facteurs portant sur des critères stratégiques de production, par exemple, on en améliorerait son efficacité de sélection, tout en renforçant son aspect processus.

D'autre part, afin de mieux intégrer l'utilisateur dans l'évaluation des processus d'assemblage, il devrait exister la possibilité de modifier ou de choisir la fonction coût selon les besoins ou les désirs de l'expert. Ceci est possible, par exemple, en augmentant le nombre et le type de paramètres d'évaluation, et en introduisant des facteurs de pondération. L'utilisateur peut alors construire sa fonction coût en les combinant à sa guise.

L'approche multi-agents offre la flexibilité (non encore totalement exploitée) et la robustesse nécessaires au changement dynamique des données et des solutions, ce qui est indispensable pour explorer l'espace tout en s'adaptant aux exigences de l'utilisateur.

Par contre, l'utilisation d'un système multi-agents dans le cas du module de sélection, ne constitue pas forcément une bonne approche d'un point de vue performance: l'espace exploré étant bien défini et le seul moyen de contrôle de l'exploration étant constitué de la fonction coût, une approche "classique" de l'optimisation est dans tous les cas plus efficace.

10.4 Module de conception

Le module de conception a pour but d'explorer l'espace des flots d'assemblage admissibles à la recherche de l'îlot le moins cher.

L'entrée du module de conception est un processus opératoire déterminé sur la base du processus d'assemblage sélectionné par le module de sélection.

Les résultats fournis par le module de conception sont:

- soit les différents flots d'assemblage formant la trajectoire de l'exploration de l'espace de recherche
- soit directement l'îlot d'assemblage optimal résultant d'un parcours de l'espace de recherche

Ceci signifie que l'utilisateur peut soit suivre pas à pas l'exploration de l'espace de recherche, soit obtenir directement un îlot d'assemblage optimal.

Dans le module de conception, l'utilisateur peut influencer la génération des flots d'assemblage en imposant le choix d'équipement pour certaines opérations. Il peut de cette manière également pallier aux carences du système.

10.4.1 Exemple

Les figures 10.19 à 10.22 donnent les descriptions d'îlots d'assemblage déterminés par le système pour un temps de cycle contrat de 5 secondes, sur la base du processus d'assemblage optimal Proc-04 (cf. tableau 10.1).

Les tâches que réalisent les îlots d'assemblage sont celles du processus opératoire formé des tâches du processus d'assemblage optimal, complétées par des tâches auxiliaires: chargement (loading-3), déchargement (unloading-1) et réorientation (reorientation-2, reorientation-4). Le processus opératoire est représenté figure 10.18.

Chaque élément de l'îlot (îlot, cellule, poste, équipement) est décrit par son nom, suivi de son temps de cycle et de son coût. Dans la première colonne, on trouve l'îlot et l'équipement de transfert reliant ses cellules. La deuxième colonne donne les cellules et, dans le cas des cellules multi-postes, l'équipement de transfert reliant les postes qui les composent. De la troisième à la septième colonne, on trouve la description des postes et, par catégorie, la description des équipements qui les composent. La huitième colonne donne les tâches exécutées dans le poste.

A la lecture des figures, nous pouvons constater les faits suivants:

- les îlots d'assemblage se distinguent par une structure différente (constitution des postes et des cellules) qui fixent alors de manière plus ou moins rigide les équipements possibles: pour un poste réalisant les mêmes tâches dans deux îlots différents, on retrouve les mêmes équipements (type et nombre), tels que les postes Station-5 de l'îlot Assembly Line-1 (figure 10.19) et Station-5 de l'îlot Assembly Line-2 (figure 10.20), ou encore les postes Station-2 de l'îlot Assembly Line-3 (figure 10.21) et Station-2 de l'îlot Assembly Line-4 (figure 10.22). Cela provient de la dynamique des agents qui les poussent à modifier leur organisation, plutôt qu'à modifier leur structure interne;
- la différence de coût entre îlots n'est pas élevée. La source principale en est justement les différences structurelles avec pour conséquence que, dans les exemples présentés, la différence entre l'îlot le moins cher et le plus cher, ne représente que 6 pour cent du coût de l'îlot le moins cher;
- le temps de cycle des îlots ne vaut pas 5 secondes. Cela provient de la définition de la condition de satisfaction technologique des agents-postes qui les forcent à minimiser la différence entre leur temps de cycle et le temps de cycle contrat. Cette différence, selon les circonstances, peut ne pas être nulle;
- la détermination des transferts formant les liens entre postes et entre cellules est un peu sommaire. Elle devrait être complétée par la détermination des postes de chargement et de déchargement des cellules (si nécessaire) et une meilleure définition du flux des constituants dans l'installation;

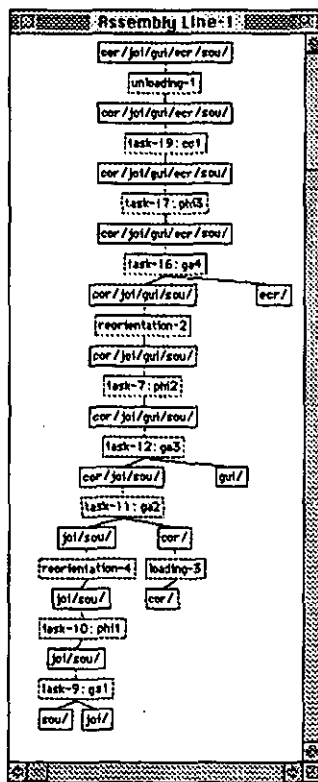


Figure 10.18: Processus opératoire réalisé dans les flots d'assemblage

Assembly Line 2	3.1	3.2	3.3	11.1.7	Cellule
Transfer					
Conveyor 1	0	19	CB2.7		
			Transfer		
			Poste		
			Station 2	3.5	11.1.2
			Equip. N°		
			Précise 1	0	3.2
			Viewing beam 1	0	3.2
			Viewing beam 2	0	3.2
			Operations	0	3.2
			Robot 1	0	3.2
			Robot 2	0	3.2
			Précise 2	0	3.2
			Viewing beam 3	0	3.2
			Viewing beam 4	0	3.2
			Operations	0	3.2
			Robot 3	0	3.2
			Robot 4	0	3.2
			Précise 3	0	3.2
			Viewing beam 5	0	3.2
			Viewing beam 6	0	3.2
			Operations	0	3.2
			Robot 5	0	3.2
			Robot 6	0	3.2
			Précise 4	0	3.2
			Viewing beam 7	0	3.2
			Viewing beam 8	0	3.2
			Operations	0	3.2
			Robot 7	0	3.2
			Robot 8	0	3.2
			Précise 5	0	3.2
			Viewing beam 9	0	3.2
			Viewing beam 10	0	3.2
			Operations	0	3.2
			Robot 9	0	3.2
			Robot 10	0	3.2
			Précise 6	0	3.2
			Viewing beam 11	0	3.2
			Viewing beam 12	0	3.2
			Operations	0	3.2
			Robot 11	0	3.2
			Robot 12	0	3.2
			Précise 7	0	3.2
			Viewing beam 13	0	3.2
			Viewing beam 14	0	3.2
			Operations	0	3.2
			Robot 13	0	3.2
			Robot 14	0	3.2
			Précise 8	0	3.2
			Viewing beam 15	0	3.2
			Viewing beam 16	0	3.2
			Operations	0	3.2
			Robot 15	0	3.2
			Robot 16	0	3.2
			Précise 9	0	3.2
			Viewing beam 17	0	3.2
			Viewing beam 18	0	3.2
			Operations	0	3.2
			Robot 17	0	3.2
			Robot 18	0	3.2
			Précise 10	0	3.2
			Viewing beam 19	0	3.2
			Viewing beam 20	0	3.2
			Operations	0	3.2
			Robot 19	0	3.2
			Robot 20	0	3.2
			Précise 11	0	3.2
			Viewing beam 21	0	3.2
			Viewing beam 22	0	3.2
			Operations	0	3.2
			Robot 21	0	3.2
			Robot 22	0	3.2
			Précise 12	0	3.2
			Viewing beam 23	0	3.2
			Viewing beam 24	0	3.2
			Operations	0	3.2
			Robot 23	0	3.2
			Robot 24	0	3.2
			Précise 13	0	3.2
			Viewing beam 25	0	3.2
			Viewing beam 26	0	3.2
			Operations	0	3.2
			Robot 25	0	3.2
			Robot 26	0	3.2
			Précise 14	0	3.2
			Viewing beam 27	0	3.2
			Viewing beam 28	0	3.2
			Operations	0	3.2
			Robot 27	0	3.2
			Robot 28	0	3.2
			Précise 15	0	3.2
			Viewing beam 29	0	3.2
			Viewing beam 30	0	3.2
			Operations	0	3.2
			Robot 29	0	3.2
			Robot 30	0	3.2
			Précise 16	0	3.2
			Viewing beam 31	0	3.2
			Viewing beam 32	0	3.2
			Operations	0	3.2
			Robot 31	0	3.2
			Robot 32	0	3.2
			Précise 17	0	3.2
			Viewing beam 33	0	3.2
			Viewing beam 34	0	3.2
			Operations	0	3.2
			Robot 33	0	3.2
			Robot 34	0	3.2
			Précise 18	0	3.2
			Viewing beam 35	0	3.2
			Viewing beam 36	0	3.2
			Operations	0	3.2
			Robot 35	0	3.2
			Robot 36	0	3.2
			Précise 19	0	3.2
			Viewing beam 37	0	3.2
			Viewing beam 38	0	3.2
			Operations	0	3.2
			Robot 37	0	3.2
			Robot 38	0	3.2
			Précise 20	0	3.2
			Viewing beam 39	0	3.2
			Viewing beam 40	0	3.2
			Operations	0	3.2
			Robot 39	0	3.2
			Robot 40	0	3.2
			Précise 21	0	3.2
			Viewing beam 41	0	3.2
			Viewing beam 42	0	3.2
			Operations	0	3.2
			Robot 41	0	3.2
			Robot 42	0	3.2
			Précise 22	0	3.2
			Viewing beam 43	0	3.2
			Viewing beam 44	0	3.2
			Operations	0	3.2
			Robot 43	0	3.2
			Robot 44	0	3.2
			Précise 23	0	3.2
			Viewing beam 45	0	3.2
			Viewing beam 46	0	3.2
			Operations	0	3.2
			Robot 45	0	3.2
			Robot 46	0	3.2
			Précise 24	0	3.2
			Viewing beam 47	0	3.2
			Viewing beam 48	0	3.2
			Operations	0	3.2
			Robot 47	0	3.2
			Robot 48	0	3.2
			Précise 25	0	3.2
			Viewing beam 49	0	3.2
			Viewing beam 50	0	3.2
			Operations	0	3.2
			Robot 49	0	3.2
			Robot 50	0	3.2
			Précise 26	0	3.2
			Viewing beam 51	0	3.2
			Viewing beam 52	0	3.2
			Operations	0	3.2
			Robot 51	0	3.2
			Robot 52	0	3.2
			Précise 27	0	3.2
			Viewing beam 53	0	3.2
			Viewing beam 54	0	3.2
			Operations	0	3.2
			Robot 53	0	3.2
			Robot 54	0	3.2
			Précise 28	0	3.2
			Viewing beam 55	0	3.2
			Viewing beam 56	0	3.2
			Operations	0	3.2
			Robot 55	0	3.2
			Robot 56	0	3.2
			Précise 29	0	3.2
			Viewing beam 57	0	3.2
			Viewing beam 58	0	3.2
			Operations	0	3.2
			Robot 57	0	3.2
			Robot 58	0	3.2
			Précise 30	0	3.2
			Viewing beam 59	0	3.2
			Viewing beam 60	0	3.2
			Operations	0	3.2
			Robot 59	0	3.2
			Robot 60	0	3.2
			Précise 31	0	3.2
			Viewing beam 61	0	3.2
			Viewing beam 62	0	3.2
			Operations	0	3.2
			Robot 61	0	3.2
			Robot 62	0	3.2
			Précise 32	0	3.2
			Viewing beam 63	0	3.2
			Viewing beam 64	0	3.2
			Operations	0	3.2
			Robot 63	0	3.2
			Robot 64	0	3.2
			Précise 33	0	3.2
			Viewing beam 65	0	3.2
			Viewing beam 66	0	3.2
			Operations	0	3.2
			Robot 65	0	3.2
			Robot 66	0	3.2
			Précise 34	0	3.2
			Viewing beam 67	0	3.2
			Viewing beam 68	0	3.2
			Operations	0	3.2
			Robot 67	0	3.2
			Robot 68	0	3.2
			Précise 35	0	3.2
			Viewing beam 69	0	3.2
			Viewing beam 70	0	3.2
			Operations	0	3.2
			Robot 69	0	3.2
			Robot 70	0	3.2
			Précise 36	0	3.2
			Viewing beam 71	0	3.2
			Viewing beam 72	0	3.2
			Operations	0	3.2
			Robot 71	0	3.2
			Robot 72	0	3.2
			Précise 37	0	3.2
			Viewing beam 73	0	3.2
			Viewing beam 74	0	3.2
			Operations	0	3.2
			Robot 73	0	3.2
			Robot 74	0	3.2
			Précise 38	0	3.2
			Viewing beam 75	0	3.2
			Viewing beam 76	0	3.2
			Operations	0	3.2
			Robot 75	0	3.2
			Robot 76	0	3.2
			Précise 39	0	3.2
			Viewing beam 77	0	3.2
			Viewing beam 78	0	3.2
			Operations	0	3.2
			Robot 77	0	3.2
			Robot 78	0	3.2
			Précise 40	0	3.2
			Viewing beam 79	0	3.2
			Viewing beam 80	0	3.2
			Operations	0	3.2
			Robot 79	0	3.2
			Robot 80	0	3.2
			Précise 41	0	3.2
			Viewing beam 81	0	3.2
			Viewing beam 82	0	3.2
			Operations	0	3.2
			Robot 81	0	3.2
			Robot 82	0	3.2

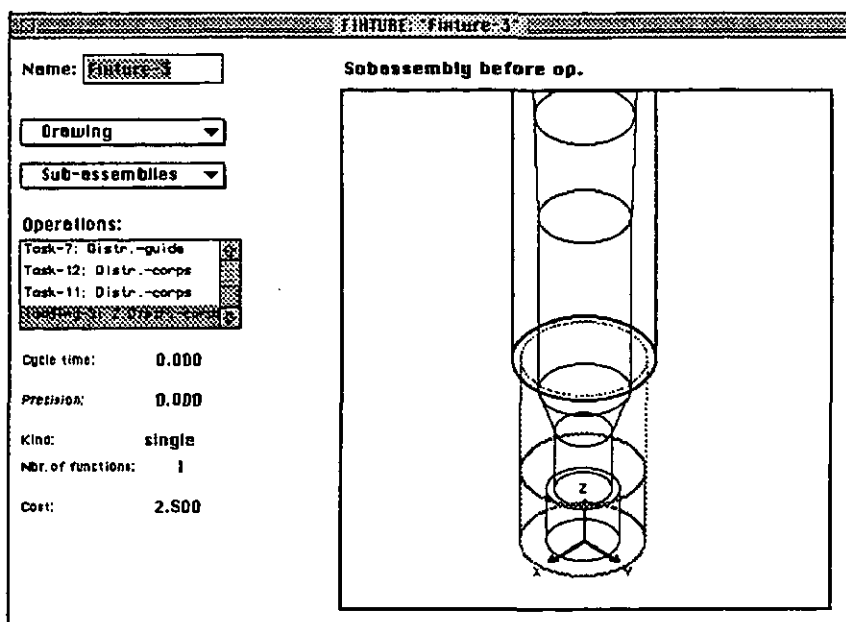


Figure 10.23: Fenêtre de visualisation d'un posage NO avec les opérations qui lui sont attribuées. Les faces du corps en grisé constituent des suggestions pour le maintien de la pièce dans le posage, basées sur la précision

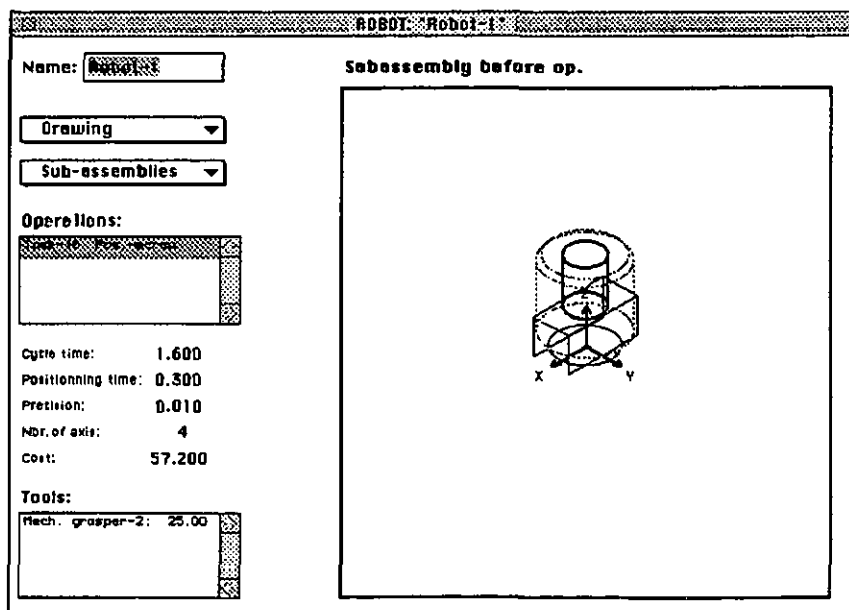


Figure 10.24: Fenêtre de visualisation d'un robot avec les opérations qui lui sont attribuées.

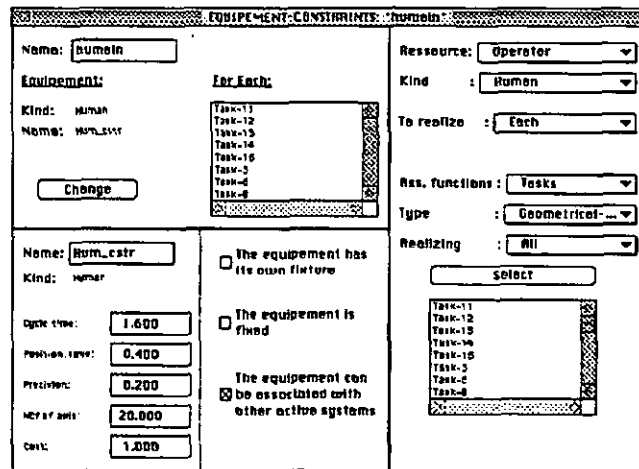


Figure 10.25: Fenêtre d'édition des contraintes-utilisateur sur le choix des équipements

- les configurations des postes et des équipements, issues des règles de construction implantées dans les agents-postes⁶, ne sont pas idéales: plusieurs posages au niveau NO, plusieurs préhenseurs sur un même opérateur et plusieurs systèmes actifs dans un même poste. L'utilisation directe de ces configurations dans un îlot réel représenterait un effort important de développement et de mise au point, pour un résultat peu efficace;
- les charges par poste ne sont pas équilibrées, ce qui constitue du point de vue de la gestion de production une erreur. Ceci provient des règles de construction des postes et de la dynamique des agents-postes. On pourrait envisager de modifier cette dynamique en créant une seconde phase dans la génération des îlots d'assemblage, durant laquelle les agents-postes tenteraient de remédier à ces disproportions.

Les résultats fournis par le module de conception ne sont pas que la description des îlots d'assemblage, mais aussi une description de chacun des équipements qui les constituent. Cette description comporte le temps de cycle de l'équipement, son coût, les opérations pour lesquelles il est prévu et, selon le type d'équipement, des suggestions sur les faces à utiliser pour la manipulation des sous-assemblages impliqués. Les figures 10.23 et 10.24 donnent des exemples de fenêtres de visualisation pour un posage et un robot.

Les contraintes-utilisateur

Dans le module de conception, l'utilisateur a la possibilité de définir des contraintes sur le choix des équipements pour la réalisation des tâches d'assemblage. Les agents concernés par ces contraintes s'adaptent alors à ces choix de l'utilisateur. La figure 10.25 montre la fenêtre d'édition pour ce genre de contraintes et la figure 10.26 décrit un îlot déterminé en tenant compte de la contrainte-utilisateur d'utiliser un humain comme opérateur, pour la réalisation de chacune des tâches géométriques.

⁶Cf. annexe D.2.

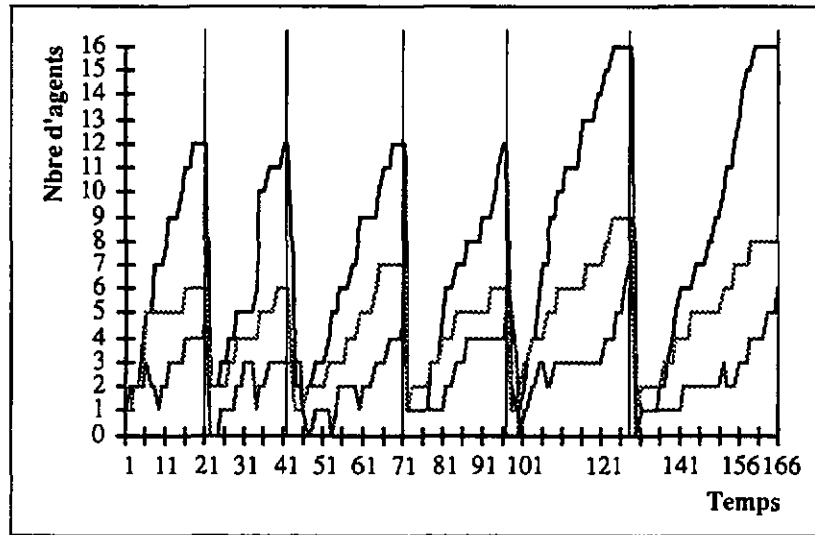


Figure 10.27: Dynamique des populations d'agents au cours du temps: en noir, nombre d'agents-tâches satisfaits; en gris clair, nombre d'agents-postes; en gris foncé, nombre d'agents-cellules

L'expression de ces contraintes devrait pouvoir porter non seulement sur le choix des équipements, mais aussi sur les règles de construction des postes, cellules et flots, afin de mieux intégrer l'utilisateur dans le processus de conception, en tant qu'agent à part entière du système.

La dynamique des agents

La figure 10.27 montre l'évolution des populations d'agents au cours du temps, pour la génération de 6 flots d'assemblage successifs. Les 4 premiers flots ont été conçus sur la base du processus opératoire issu du processus d'assemblage optimal et les 2 derniers sur la base du processus opératoire issu du processus d'assemblage le plus mauvais.

Comme pour le module de sélection, l'échelle de temps n'est pas linéaire, mais proportionnelle au nombre de modifications qu'a subies l'flot d'assemblage au cours de sa génération: les mesures de ces populations ont été faites par l'agent-ligne, chaque fois qu'il recevait un avis de modification ou de décès de la part d'un agent-cellule. L'ordre de grandeur temporel pour la génération d'un flot d'assemblage est d'environ 1 minute sur un PowerPC à 166Mhz avec 24 MB de RAM.

On constate les faits suivants:

- pour les mêmes raisons que dans le module de sélection, le temps nécessaire pour générer un flot d'assemblage n'est pas constant.
- les populations des agents-tâches satisfaits et des agents-postes ne font que croître au cours de la génération d'un flot, alors que la population d'agents-cellules peut subir des variations. Cela provient du fait que les agents-postes acceptent un maximum d'agents-tâches (tant que les contraintes de

précédence le permettent) avant de se bloquer, ce qui signifie qu'il n'y a pas de suicide d'agents-postes au cours de la génération d'un flot, alors que les agents-cellules commencent par rejeter des agents-postes déjà placés avant d'accepter un autre agent-poste, ce qui peut amener à un suicide.

L'utilisation d'un système multi-agents pour le module de conception semble répondre naturellement au problème: du fait de la localité intrinsèque des règles de conception qui définissent une structure donnée (équipement, poste, cellule, flot) et ses relations avec les autres structures, leur encapsulation dans des agents facilite leur utilisation et permet ainsi de mieux maîtriser la conception.

De plus, le système offre une souplesse et une robustesse face aux changements dynamiques des contraintes que n'offre peut-être pas une approche plus rigide.

Par contre, comme pour le module de sélection, la dynamique multi-agent semble ne pas apporter de gain de temps.

10.4.2 Evaluation

Les exemples d'îlots d'assemblage générés par le module de conception ont montré que:

- les règles de construction d'un flot d'assemblage implantées actuellement dans le comportement des agents ne sont pas des plus représentatives de l'expertise du domaine, ce qui fait que l'aspect des flots d'assemblage n'est pas technologiquement satisfaisant. Ces règles s'appuient principalement sur des considérations de précédence pour l'acceptation et le rejet des agents. En s'appuyant plus sur des critères technologiques (par exemple, le regroupement possible ou non de tâches dans un équipement), on pourrait améliorer les résultats;
- les flots générés se distinguent plus par leur organisation de postes et de cellules que par la structure interne de leurs postes, avec pour conséquence une disparité des charges par poste. Pour remédier à cela, on pourrait modifier la dynamique des agents-postes pour les forcer à améliorer leur structure interne, en introduisant une seconde condition de satisfaction technologique (et de nouveaux comportements associés). Celle-ci serait non plus basée sur le temps de cycle (ce qui fixe sa taille), mais sur la qualité technologique de sa structure interne au moyen de nouvelles règles telles celles décrites au premier point;
- l'estimation du coût de l'îlot, telle qu'elle est calculée par le système, donne des résultats proportionnellement corrects, bien que non relatifs au coût réel d'un îlot d'assemblage.

Ceci implique que les règles de conception devrait être modifiées, voire que l'utilisateur puisse lui-même introduire ses propres règles, en définissant les conditions de satisfaction des agents, par exemple. Une autre possibilité serait d'offrir à l'utilisateur une palette de conditions de satisfaction (basées sur le temps de cycle, la nature des sous-assemblages impliqués, le type d'équipement à disposition, le coût, etc.) et leurs comportements associés pour chaque agent, afin qu'il construise son propre système multi-agents de conception, selon le problème à résoudre, son expérience et ses désirs.

Dans son implantation actuelle, l'interface servant à représenter les flots d'assemblage résultants n'est pas des plus explicite et devrait être améliorée.

En ce qui concerne l'approche multi-agents dans le cas du module de conception, elle offre pleinement (bien que non encore totalement exploité) l'avantage de sa flexibilité face à la modification dynamique des contraintes par l'utilisateur. Celui-ci peut intervenir en imposant des contraintes sur la réalisation des tâches, les agents s'adaptant à ces exigences. Cet atout, par rapport à une approche "classique", devrait être encore amélioré en exploitant mieux les capacités de l'expert, par son intervention directe sur le comportement des agents.

10.5 Conclusion

Le système AMACOIA a été conçu dans l'optique de s'adapter aux exigences et à l'expertise de l'utilisateur, en se basant sur le principe selon lequel les capacités d'analyse d'un expert humain sont supérieures (en qualité et en temps) à celles de l'ordinateur, alors que les capacités de calcul de l'ordinateur sont supérieures à celles de l'expert humain.

Pour ce faire, le système s'appuie sur l'arbre des références opératoires et le graphe de précedence qui permettent à l'utilisateur de décrire le sous-espace de l'espace fonctionnel qu'il souhaite explorer, et les systèmes multi-agents dont la souplesse et la robustesse s'adaptent parfaitement à un changement dynamique des données.

Le traitement de l'exemple de la valve de vélo a permis de démontrer la validité de l'approche, mais a aussi montré que, dans son état actuel, la capacité d'adaptation du système n'est pas totalement exploitée: il manque encore des outils à disposition de l'utilisateur, pour qu'il puisse mieux s'intégrer au processus de conception.

D'autre part, les paramètres de la fonction coût du module de sélection (à l'exception du facteur d'utilisation des récepteurs) semblent mal adaptés à la sélection de processus d'assemblage. Sous cette forme, ces paramètres semblent mieux convenir pour évaluer des variantes de conception d'un produit, du point de vue de l'assemblage.

De même, les règles de conception des flots d'assemblage devraient être étendues, afin de prendre en compte plus d'éléments importants pour l'assemblage (les tolérances, les critères stratégiques, la gestion de production, par exemple) qu'actuellement. En effet la qualité des solutions trouvées n'est pas des meilleures, faute d'une base de règles suffisante.

La correction de ces imperfections (qui n'enlèvent rien à la validité de l'approche), devrait permettre à AMACOIA de remplir son but: être un outil simple et adaptatif, support de la créativité de l'ingénieur.

Une autre extension possible du système serait aussi de le transformer pour qu'il soit utilisé par les concepteurs de produit, dans le but d'évaluer le coût d'assemblage de leur produit. Ceci signifierait une abstraction supplémentaire des données d'entrées, débouchant sur une simplification de la saisie des données et une prise en charge par le système de la compétence en assemblage.

Chapitre 11

Conclusion

La nature même de la conception ne permet pas son traitement automatique par des moyens informatiques. Par contre il s'avère possible de l'appréhender au travers de modèles, manipulables informatiquement, sur la base desquels il est alors possible de structurer et d'évaluer les étapes du processus de conception.

De plus, il est de plus en plus important pour un concepteur de pouvoir analyser rapidement les conséquences de ses choix afin d'en minimiser l'impact économique.

Le travail présenté dans ce mémoire s'inscrit dans cette idée en offrant un *outil d'aide* à la conception des systèmes d'assemblage aussi bien pour la détermination et la sélection des processus d'assemblage que pour la caractérisation des équipements. L'expert humain peut donc configurer le système, selon les capacités de raisonnement d'AMACOIA, la difficulté et les contraintes du problème à résoudre et le degré de précision souhaité pour les résultats. L'utilisateur fixant ainsi la répartition des tâches entre lui et l'ordinateur, le système s'adapte alors aux conditions imposées par l'humain.

Preuant pour point de départ la formalisation développée pour les problèmes de conception en intelligence artificielle, nous avons présenté la conception des systèmes d'assemblage comme un parcours à travers deux espaces de recherche:

- l'espace fonctionnel, constitué des processus d'assemblage admissibles pour le produit, dont l'optimum est le processus d'assemblage dont les tâches sont les plus simples à réaliser;
- et l'espace physique, constitué des systèmes d'assemblage qui réalisent ces processus, dont l'optimum est le système d'assemblage au coût technologique le moins élevé.

Le traitement de ces espaces est effectué par des systèmes multi-agents auxquels appartient aussi l'expert humain. Ces systèmes multi-agents sont basés sur le modèle de l'éco-résolution proposé par J. Ferber [25]. Ce modèle a été étendu, en y introduisant la notion de condition d'existence pour les agents: des agents apparaissent et disparaissent dynamiquement, ce qui augmente la flexibilité du système. La dynamique de ces SMA permet à la fois la génération des espaces et leur parcours sous contrôle du recuit simulé. Cette approche offre les avantages suivants:

- les systèmes multi-agents permettent de gérer la complexité des interdépendances entre les différentes contraintes du problème: chaque agent est spécialisé dans la résolution d'un sous-problème spécifique, et les conflits se règlent par interactions entre les agents;
- les systèmes multi-agents offrent une capacité d'adaptation aux perturbations: si l'un des agents disparaît ou est modifié, ou encore si certaines contraintes sont relâchées ou renforcées, le système va tendre automatiquement vers une nouvelle solution;
- le recuit simulé permet la recherche d'une solution optimale en évitant le piège des optima locaux.

De plus, du point de vue de l'utilisateur, les systèmes multi-agents offrent une aide efficace pour structurer et contrôler le processus de conception, puisqu'ils découpent déjà le problème en sous-problèmes (chaque agent est spécialisé dans un sous-problème spécifique) et qu'ils offrent les mécanismes d'interactions entre ces sous-problèmes.

Parallèlement au développement des systèmes multi-agents, nous avons développé un nouveau modèle du produit, qui est *orienté processus*, basé sur le modèle opératoire de J. M. Henrioud [40]. Ce modèle, appelé l'arbre des références opératoires, ne regroupe que les informations pertinentes à l'assemblage du produit: caractéristiques à réaliser au cours du processus, modèle géométrique des constituants orienté assemblage, choix technologique pour la réalisation des caractéristiques.

Ce modèle, associé à un graphe de précédence, permet de réduire la taille de l'espace fonctionnel, tout en garantissant de rester dans l'espace des solutions possibles.

L'implantation d'un prototype et son test sur des cas industriels ont prouvé la validité de l'approche, mais certains points devraient être encore développés pour en améliorer l'efficacité:

- les données géométriques introduites par l'utilisateur ne sont pas pleinement utilisées par le système, ou incomplètes:
 - les tolérances ne sont utilisées que pour sélectionner les faces de positionnement et de saisie possibles, alors qu'elles jouent un rôle non négligeable dans le choix des équipements;
 - certains éléments géométriques des pièces qui ont une importance en assemblage (chanfreins, états de surface, rigidité des pièces), ne sont pas décrits et ne sont pas pris en compte dans le processus de conception.

Ceci a pour conséquence un appauvrissement de la qualité des solutions proposées.

De plus, seul le cas mono-produit a été abordé. L'extension de la modélisation au cas multi-produits permettrait de mieux satisfaire la réalité du marché actuel.

- l'interface de saisie de l'arbre des références opératoires devrait être améliorée: la méthode de description de la géométrie est un peu fastidieuse.

Pour la simplifier, on pourrait envisager l'utilisation d'un système de DAO couplé à AMACOIA, bien que dans la pratique les concepteurs de systèmes d'assemblage n'ont encore que le papier comme source principale d'information.

- la fonction d'évaluation de la difficulté d'exécution des tâches d'assemblage montre quelques faiblesses dans la caractérisation des processus d'assemblage:
 - les facteurs de stabilité, d'accessibilité et de parallélisation permettent davantage d'évaluer l'assemblabilité d'un produit, plutôt que de caractériser des processus d'assemblage. Il faudrait donc modifier leur calcul afin de mieux intégrer l'aspect processus, par exemple en se basant plutôt sur les sous-assemblages concernés que sur les caractéristiques d'assemblage du produit.
 - elle ne porte que sur la difficulté de réalisation des tâches d'assemblage. En introduisant dans la fonction de nouveaux facteurs portant sur des critères stratégiques de production, par exemple, on en améliorerait son efficacité de sélection, tout en renforçant son aspect processus.
- les règles de construction d'un flot d'assemblage implantées actuellement dans le comportement des agents ne sont pas des plus représentatives de l'expertise du domaine, ce qui fait que l'aspect des flots d'assemblage n'est pas technologiquement satisfaisant. De plus, l'équilibrage des charges sur les postes n'est pas correct (trop grande disparité de temps de cycle). La dynamique du système multi-agents devrait être modifiée afin de tenir compte de ces aspects.
- l'expert humain reste encore un agent assez passif. En augmentant les moyens d'interactions entre l'expert et AMACOIA, la dynamique du système pourrait être améliorée:
 - le découpage fonctionnel des processus d'assemblage, effectué uniquement par le système devrait pouvoir être fait interactivement avec l'utilisateur, afin de s'adapter aux cas non triviaux et d'améliorer les performances des modules de sélection et de conception.
 - il devrait exister la possibilité de modifier ou de choisir la fonction d'évaluation selon les besoins ou les désirs de l'expert.
 - dans le module de conception, l'utilisateur peut intervenir en imposant des contraintes sur la réalisation des tâches, les agents s'adaptant à ces exigences. Cet atout, par rapport à une approche "classique", devrait être encore amélioré en exploitant mieux les capacités de l'expert, par son intervention directe sur le comportement des agents: "programmation" des agents, expression des préférences de l'expert pour les résultats d'un des agents, etc.

Le développement de ces quelques points ferait du système AMACOIA un outil vraiment efficace pour les concepteurs de systèmes d'assemblage, mais aussi pour les concepteurs de produit afin qu'ils évaluent les conséquences de leur choix de conception sur le coût de l'assemblage.

Bibliographie

- [1] Emile Aarts and Jan Korst. *Simulated annealing and Boltzmann Machines*. John Wiley and Sons, 1989.
- [2] Eric Bonado and Guy Théraulaz, editors. *Intelligence Collective*. Hermes, 1995.
- [3] Alain Bourjault. *Contribution à une approche méthodologique de l'assemblage automatisé: élaboration automatique des séquences opératoires*. PhD thesis, Université de Franche-Comté, Besançon, France, 1984.
- [4] Jean-Paul Bourrières. *Contribution à la modélisation intégrée des systèmes robotisés d'assemblage*. PhD thesis, Université de Franche-Comté, Besançon, France, 1990.
- [5] Forrest D. Brewer and Daniel D. Gajski. Knowledge-based control in micro-architecture design. In *Proceedings of the 24th Design Automation Conference*, pages 203–209, 1987.
- [6] Axel Brinkop, Norbert Laudwein, and Rüdiger Maasen. Routine design for mechanical engineering. *AI Magazine*, 16(1): 74–85, 1995.
- [7] David C. Brown and B. Chandrasekaran. Expert systems for a class of mechanical design activity. In John S. Gero, editor, *Knowledge Engineering in Computer-Aided Design: proceedings of the IFIP WG 5.2 Working Conference on Knowledge Engineering in Computer-Aided Design, Budapest, 1984*, pages 259–282. North-Holland, 1985.
- [8] Cristiano Castelfranchi and Jean-Pierre Müller, editors. *From Reaction to Cognition - Selected Papers of the 5th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'93*. Springer-Verlag, 1995.
- [9] B. Chandrasekaran. Design problem solving: a task analysis. *AI Magazine*, 11(4): 59–71, 1990.
- [10] D. Chappe and A. Bourjault. Utilisation des réseaux de petri temporisés pour la conception et l'évaluation des systèmes d'assemblage automatisés. In *Proceedings of the 12th IMACS World Congress on Scientific Computation*, 1988.
- [11] We-Min Chow. *Assembly line design: Methodoly and Applications*. Marcel Dekker, 1990.

- [12] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42, 1990.
- [13] Roberto Cremonini, Evelina Lamma, and Paola Mello. Ades: an expert system for atp design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 3(1): 1-21, 1989.
- [14] Mark R. Cutkosky, Robert S. Engelmores, Richard E. Fikes, Michael R. Genesereth, Thomas R. Gruber, William S. Mark, Jay M. Tenenbaum, and Jay C. Weber. Pact: an experiment in integrating concurrent engineering systems. *IEEE Computer Magazine*, 26(1): 28-37, 1993.
- [15] Kishor P. Dabke, Le Nguyen Binh, and Sien Peng Lau. Expert system for the design of optical fibre communications links. *Engineering Applications of Artificial Intelligence*, 8(3): 319-331, 1995.
- [16] Thouraya Daouas. *Ordonnancement distribué réactif pour un atelier d'assemblage de type flow shop*. PhD thesis, Université de Neuchâtel, Suisse, 1996.
- [17] Thouraya Daouas, Khaled Ghedira, and Jean-Pierre Müller. How to schedule a flow shop plant by agents. In *Proceedings of the 10th International Conference on Engineering Applications of Artificial Intelligence, AIENG'95*, 1995.
- [18] D. de Werra and A. Hertz. Tabu search techniques: a tutorial and an application to neural networks. *OR Spektrum*, 11: 131-141, 1989.
- [19] Yves Demazeau and Jean-Pierre Müller, editors. *Decentralized Artificial Intelligence*. North Holland, 1990.
- [20] Yves Demazeau and Jean-Pierre Müller, editors. *Decentralized Artificial Intelligence vol. 2*. North Holland, 1991.
- [21] John R. Dixon. Artificial intelligence and design: a mechanical engineering view. In AAAI/Morgan Kaufmann, editor, *Proceedings of the 5th National Conference on Artificial Intelligence*, pages 872-877, 1986.
- [22] Najah Djemel. *Contribution à la conception des systèmes flexibles d'assemblage dans le cas multi-produits*. PhD thesis, Université de Franche-Comté, Besançon, France, 1994.
- [23] Alexis Drogoul. When ants play chess (or can strategies emerge from tactical behaviour?). In Cristiano Castelfranchi and Jean-Pierre Müller, editors, *From Reaction to Cognition - Selected Papers of the 5th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAA-MAW'99*, pages 13-27. Springer-Verlag, 1995.
- [24] Alexis Drogoul and Christophe Dubreuil. Eco-problem solving: Results of the n-puzzle. In Eric Werner and Yves Demazeau, editors, *Decentralized Artificial Intelligence vol. 3*, pages 283-295. North Holland, 1992.
- [25] Jacques Ferber. *Objets et agents: une étude des structures de représentation et de communication en Intelligence Artificielle*. PhD thesis, Université de Paris VI, France, 1989.

- [26] Jean Figour. Techniques d'assemblage 1 - spécificité de l'assemblage. Technical report, École Polytechnique Fédérale de Lausanne, Lausanne, Suisse, 1989. cours.
- [27] R. Fletcher. *Practical Methods of Optimization - Second Edition*. John Wiley and Sons, 1987.
- [28] Stephanie Forrest. Emergent computation: Self-organizing, collective, and cooperative phenomena in natural and artificial computing networks. *Physica*, 42: 1–11, 1990.
- [29] Graça Gaspar. Communication and belief changes in a society of agents: towards a formal model of an autonomous agent. In Yves Demazeau and Jean-Pierre Müller, editors, *Decentralized Artificial Intelligence vol. 2*, pages 245–255. North Holland, 1991.
- [30] John S. Gero. Design prototypes: a knowledge representation schema for design. *AI Magazine*, 11(4): 26–36, 1990.
- [31] Khaled Ghedira. *MASC: une approche multi-agents des problèmes de satisfaction de contraintes*. PhD thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France, 1993.
- [32] F. Glover. Tabu search - part 1. *ORSA Journal on Computing*, 1: 190–206, 1989.
- [33] F. Glover. Tabu search - part 2. *ORSA Journal on Computing*, 2: 4–32, 1990.
- [34] Piotr J. Gmytrasiewicz, H. H. Huang, and Frank L. Lewis. Combining operations research and agent-oriented techniques for agile manufacturing system design. In *Proceedings of the Workshop on Intelligent Manufacturing Systems of the IJCAI'95*, pages 124–135, 1995.
- [35] Vinod Goel and Peter Pirolli. Motivating the notion of generic design within information-processing theory: the design problem space. *AI Magazine*, 10(1): 18–36, 1989.
- [36] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
- [37] Deborah M. Gordon. The organization of work in social insect colonies. *Nature*, 380(6570): 121–124, 1996.
- [38] Feliz Ribeiro Gouveia and Eugénio da Costa Oliveira. Agent-oriented modelling of manufacturing systems. In *Proceedings of the Workshop on Intelligent Manufacturing Systems of the IJCAI'95*, pages 136–145, 1995.
- [39] Daniel R. Greening. Parallel simulated annealing techniques. *Physica*, 42: 293–306, 1990.
- [40] Jean-Michel Henrioud. *Contribution à la conceptualisation de l'assemblage automatisé: nouvelle approche en vue de la détermination des processus d'assemblage*. PhD thesis, Université de Franche-Comté, Besançon, France, 1989.

- [41] Jean-Michel Henrioud and Alain Bourjault. Lega: a computer-aided generator of assembly plans. In Luiz S. Homem de Mello and Shukan Lee, editors, *Computer-Aided Mechanical Assembly Planning*, pages 191–215, Boston, Dordrecht, London, 1991. Kluwer Academic Publishers.
- [42] Luiz S. Homem de Mello and Arthur C. Sanderson. A basic algorithm for the generation of mechanical assembly sequences. In Luiz S. Homem de Mello and Shukan Lee, editors, *Computer-Aided Mechanical Assembly Planning*, pages 163–190. Kluwer Academic Publishers, 1991.
- [43] Luiz S. Homem de Mello and Arthur C. Sanderson. Representations for assembly sequences. In Luiz S. Homem de Mello and Shukan Lee, editors, *Computer-Aided Mechanical Assembly Planning*, pages 130–162, Boston, Dordrecht, London, 1991. Kluwer Academic Publishers.
- [44] Max-Olivier Hongler. Etude analytique du comportement dynamique d'une chaîne de production. Technical report, Ecole Polytechnique Fédérale de Lausanne, Suisse, 1992.
- [45] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. Technical report, IBM Research Report RC 9355, 1982.
- [46] Daniel Kuokka and Brian Livezey. A collaborative parametric design agent. In AAAI Press/ the MIT Press, editor, *Proceedings of the 12th National Conference on Artificial Intelligence*, pages 387–393, 1994.
- [47] Mary Lou Maher. Process models for design synthesis. *AI Magazine*, 11(4): 49–58, 1990.
- [48] Christian Mascle. *Approche méthodologique de détermination de gammes par le désassemblage*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Suisse, 1990.
- [49] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1994.
- [50] Sanjay Mittal, Clive L. Dym, and Mahesh Morjaria. Pride: an expert system for the design of paper handling systems. *IEEE Computer*, pages 102–114, juillet 1986.
- [51] P. Olivier. *Analyse et synthèse des systèmes d'assemblage automatisés*. PhD thesis, Université de Franche-Comté, Besançon, France, 1986.
- [52] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, 1982.
- [53] Christophe Ferrard. *Contribution à une méthodologie d'aide à l'implantation et à la spécification des équipements des systèmes flexibles d'assemblage*. PhD thesis, Université de Franche-Comté, Besançon, France, 1992.
- [54] Samuel Pierre, Michel-Ange Hyppolite, Jean-Marie Bourjolly, and Oumar Dioume. Topological design of computer communication networks using simulated annealing. *Engineering Applications of Artificial Intelligence*, 8(1): 61–69, 1995.

- [55] Jean-Claude Prélaz. Conception d'installations d'assemblage. notes personnelles, 1989.
- [56] R. Bharat Rao and Stephen C-Y. Lu. Building models to support synthesis in early stage product design. In *Proceedings of the 11th National Conference on Artificial Intelligence, AAAI'93*, pages 277–282, 1993.
- [57] Paul Robert and al. *Le Petit Robert 1 - Dictionnaire alphabétique et analogique de la langue française*. Les Dictionnaires Le Robert, 1985.
- [58] M. D. Rychener, M. L. Farinacci, I. Hulthage, and M. S. Fox. Integration of multiple knowledge sources in aladin, an alloy design system. In AAAI/Morgan Kaufmann, editor, *Proceedings of the 5th National Conference on Artificial Intelligence*, pages 878–882, 1986.
- [59] Arthur C. Sanderson, Luiz S. Homem de Mello, and Hui Zhang. Assembly sequence planning. *AI Magazine*, 11(1): 62–81, 1990.
- [60] J. R. Searle. *Speech acts*. Cambridge University Press, 1969.
- [61] Yoav Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1): 51–92, 1993.
- [62] François Sprumont and Jean-Pierre Müller. Kasba: a knowledge based system for flexible assembly execution. In *Proceedings of the 8th International Conference on Engineering Applications of Artificial Intelligence, AIENG'93*, pages 813–826, 1993.
- [63] François Sprumont and Jean-Pierre Müller. Amacoia: a multi-agent system for designing flexible assembly lines. In *Proceedings of the 1st International Conference on Practical Application of Intelligent Agents and Multi-Agent Technology, PAAM'96*, pages 573–585, 1996.
- [64] François Sprumont, Jean-Pierre Müller, and Khaled Ghedira. Amacoia: a multi-agent approach to the design of flexible assembly lines - preliminary report. In *Proceedings of the Workshop on Intelligent Manufacturing Systems of the IJCAI'95*, 1995.
- [65] Christopher Tong and Duvvuru Sriram. *Artificial Intelligence in Engineering Design Vol 1 - Design Representation and Models of Routine Design*, chapter Introduction. Academic Press, 1992.
- [66] Christopher Tong and Duvvuru Sriram, editors. *Artificial Intelligence in Engineering Design Vol 1 - Design Representation and Models of Routine Design*. Academic Press, 1992.
- [67] Christopher Tong and Duvvuru Sriram, editors. *Artificial Intelligence in Engineering Design Vol 2 - Models of Innovative Design, Reasoning about Physical Systems, and Reasoning about Geometry*. Academic Press, 1992.
- [68] Christopher Tong and Duvvuru Sriram, editors. *Artificial Intelligence in Engineering Design Vol 3 - Knowledge Acquisition, Commercial Systems, and Integrated Environments*. Academic Press, 1992.

- [69] Karl T. Ulrich and Warren P. Seering. Functional sharing in mechanical design. *Design Studies*, 11(4): 223-234, 1990.
- [70] Eric Verdebout. *Pilotage algorithmique du changement de campagne de production dans une installation flexible d'assemblage automatique*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Suisse, 1992.
- [71] VSM-Normenbüro. *Extrait de normes pour écoles techniques*. Zürich, 1986.
- [72] Q. Wang, J. Y. Zhu, Y. Q. Shu, M. Rao, and K. T. Chuang. An intelligent design environment for conceptual process design. *Engineering Applications of Artificial Intelligence*, 8(2): 115-127, 1995.
- [73] Ian Watson and Farhi Marir. Case-based reasoning: a review. *The Knowledge Engineering Review*, 9(4): 327-354, 1994.
- [74] Eric Werner and Yves Demazeau, editors. *Decentralized Artificial Intelligence vol. 3*. North Holland, 1992.
- [75] D. E. Whitney, T. De Fazio, R. E. Gustavson, S. C. Graves, K. Coopriider, C. J. Klein, M. Lui, and S. Pappu. Computer-aided design of flexible assembly systems - final report. Technical Report CSDL-R-2033, The Charles Stark Draper Laboratories, Cambridge, Massachusetts, 1988.
- [76] Linda M. Wills and Janet L. Kolodner. Towards more creative case-based design systems. In AAAI Press/ the MIT Press, editor, *Proceedings of the 12th National Conference on Artificial Intelligence*, pages 50-55, 1994.
- [77] Randall H. Wilson and Jean-Claude Latombe. Geometric reasoning about mechanical assembly. *Artificial Intelligence*, 71(2): 371-396, 1994.
- [78] Julian Winsor and Kenneth Maccallum. A review of functionality modelling in design. *The Knowledge Engineering Review*, 9(2): 163-199, 1994.
- [79] Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2): 115-152, 1995.

Annexe A

Les agents du module de sélection

A.1 Les agents-tâches

A.1.1 Objectifs

Les agents-tâches du module de sélection sont générés par le système. Il existe un agent-tâche par tâche d'assemblage du graphe ET/OU, G_{AV} . Un agent-tâche n'a qu'un seul objectif:

- la tâche pour laquelle l'agent-tâche a été créé doit faire partie du processus d'assemblage sélectionné par l'agent-processus.

A.1.2 Accointances

Un agent-tâche n'a qu'une seule accointance: l'agent-processus, a_p .

A.1.3 Mémoire locale

La mémoire locale contient les connaissances de l'agent-tâche. Les connaissances d'un agent-tâche a_t^j sont décrites par le doublet:

$C^{a_t^j} = \langle t^j, cs^j \rangle$ avec:

- t^j , la tâche d'assemblage pour laquelle l'agent a été généré.
- cs^j , paramètre caractérisant la condition de satisfaction de l'agent-tâche. Il peut prendre les valeurs "Selected" ou "Not Selected" selon que t^j fasse partie ou non du processus sélectionné, ou "Sleeping" si a_t^j est désactivé.

A.1.4 Comportement

Le comportement d'un agent-tâche est de type réactif et il est guidé par la recherche de satisfaction: la réception d'un message en provenance de l'agent-processus déclenche une réaction appropriée, afin de se rapprocher de sa condition de satisfaction.

Condition de satisfaction

La condition de satisfaction d'un agent-tâche est que sa tâche d'assemblage t^j fasse partie du processus d'assemblage sélectionné par l'agent-processus ($cs^j = Selected$).

Réactions

La réaction d'un agent-tâche à un message de l'agent-processus se déroule de la manière suivante:

1. *Mise à jour de la satisfaction*: cs^j est mis à jour en fonction du message reçu de l'agent-processus (cf. tableau A.1)
2. *Envoi d'une requête*: si $cs^j = NotSelected$, il envoie une requête ("Request") à l'agent-processus, sinon il ne fait rien.

message reçu	contenu	mise à jour
<i>Accepted</i>	-	$cs^j := Selected$
<i>Rejected</i>	-	$cs^j := NotSelected$
<i>Activate</i>	-	si $cs^j \neq Selected$, $cs^j := NotSelected$
<i>Deactivate</i>	-	$cs^j := Sleeping$

Tableau A.1: Mise à jour de la satisfaction d'un agent-tâche selon le message reçu de l'agent-processus

L'acceptation et le rejet d'un agent-tâche sont liés à la sélection du processus d'assemblage. L'activation et la désactivation de l'agent-tâche sont liées au parcours de l'espace de recherche.

Information diffusée

La seule information diffusée par un agent-tâche est une demande de placement ("Request") envoyé à l'agent-processus si $cs^j = NotSelected$. Le contenu de ce message est l'agent-tâche lui-même.

A.2 L'agent-processus

A.2.1 Objectifs

Il n'y a qu'un seul agent-processus généré par le système. L'agent-processus a pour objectifs:

- la sélection d'un processus d'assemblage par combinaison des tâches fournies par les agents-tâches.
- le contrôle du parcours de l'espace de recherche par la gestion du recuit simulé sur le critère de la difficulté estimée de l'exécution des tâches ($f_d(p) = \sum_{k=1}^n d_t^k$).

A.2.2 Accointances

Les accointances de l'agent-processus sont:

- A_t , l'ensemble des agents-tâches,
- a_u , l'utilisateur du système,
- a_i , l'agent-flot.

A.2.3 Mémoire locale

La mémoire locale contient les connaissances de l'agent-processus a_p . Ces connaissances sont décrites par l'octuplet:

$C^{a_p} = \langle G_{\wedge V}, T^{a_p}, pa^{a_p}, p_{crt}^{a_p}, p_{new}^{a_p}, p_{opt}^{a_p}, A_t^{a_p}, cs^{a_p} \rangle$ avec:

- $G_{\wedge V}$, le graphe ET/OU constituant l'espace de recherche du module de sélection.
- T^{a_p} , la valeur de la tolérance au changement (température) de l'agent-processus. T^{a_p} permet aussi d'évaluer la condition de satisfaction de parcours (cf. §A.2.4).
- pa^{a_p} , la valeur du palier de tolérance de l'agent-processus.
- $p_{crt}^{a_p}$, le processus d'assemblage courant,
- $p_{new}^{a_p}$, le processus d'assemblage qui pourra constituer le nouvel état de $p_{crt}^{a_p}$,
- $p_{opt}^{a_p}$, le processus d'assemblage optimal du moment,
- $A_t^{a_p}$, l'ensemble des agents-tâches qui ont été acceptés par l'agent-processus,
- cs^{a_p} , valeur booléenne indiquant si la condition de satisfaction de cohérence est remplie ou non.

A.2.4 Comportement

Le comportement de l'agent-processus est de type réactif et il est guidé par la recherche de satisfaction: la réception d'un message en provenance d'une accointance déclenche une réaction appropriée, afin de se rapprocher de sa condition de satisfaction.

Condition de satisfaction

Les conditions de satisfaction de l'agent-processus sont:

- *Condition de satisfaction de cohérence*: l'instanciation de $p_{new}^{a_p}$ constitue un processus d'assemblage complet, amenant du pré-produit au produit fini ($cs^{a_p} = True$).
- *Condition de satisfaction de parcours*: l'agent-processus a effectué une trajectoire complète dans l'espace de recherche ($T^{a_p} = 0$).

Réactions

La réaction déclenchée par l'agent-processus dépend du message reçu et de ses conditions de satisfaction. Le tableau A.2 donne les conditions de déclenchement des réactions de l'agent-processus.

message	expéditeur	contenu	satisfaction	réaction
<i>Request</i>	a_i^k	a_i^k	$cs^{ap} = False$ et $T^{ap} > 0$ $cs^{ap} = True$ et $T^{ap} > 0$ $cs^{ap} = True$ et $T^{ap} = 0$	intégration révision désactivation
<i>New Search</i>	a_u ou a_i	-	indép. de la satisfaction	activation
<i>Process Required</i>	a_u ou a_i	<i>Current</i> ou <i>Optimal</i>	indép. de la satisfaction	information

Tableau A.2: Conditions de déclenchement des réactions de l'agent-processus, avec: a_u , l'utilisateur; a_i , l'agent-flot; a_i^k , un agent-tâche

Les réactions d'intégration et de révision de l'agent-processus sont liées à la sélection d'un processus d'assemblage. L'application du recuit simulé et les réactions de désactivation et d'activation sont, quant à elles, liées au mécanisme de parcours de l'espace de recherche. La réaction d'information permet le lien entre le module de sélection et le module de conception (cf. chapitre 9).

Réaction d'intégration La réaction d'intégration permet à l'agent-processus de se rapprocher de sa condition de satisfaction de cohérence. Elle est déclenchée par une requête de placement ("Request") en provenance d'un agent-tâche, pour peu que les conditions de satisfaction ne soient pas remplies ($cs^{ap} = False$ et $T^{ap} > 0$). Si la condition de cohérence est remplie à la suite de la réaction, l'agent-processus se rapproche de sa condition de parcours en appliquant le recuit simulé.

La réaction d'intégration se déroule de la manière suivante:

1. *Evaluation de l'agent-tâche*: L'agent-processus évalue si la tâche t^{req} de l'agent-tâche requérant a_i^{req} peut être rattachée ou non au processus partiel en construction: l'agent-tâche peut être intégré si t^{req} peut être directement rattaché à p_{new}^{ap} en respect avec les contraintes de précédence de G_{AV} . Si c'est le cas, l'agent-processus ajoute la nouvelle tâche à p_{new}^{ap} .
2. *Envoi de la décision*: Si a_i^{req} a été intégré, l'agent-processus lui envoie un message d'acceptation ("Accepted") et le rajoute à A_i^{ap} , sinon il lui envoie un message de rejet ("Rejected").
3. *Evaluation du processus d'assemblage*: Si p_{new}^{ap} constitue un processus complet, la condition de satisfaction de cohérence est remplie ($cs^{ap} = True$) et l'agent-processus applique le recuit simulé.

La probabilité d'acceptation de p_{new}^{ap} est:

$$a(T^{ap}) = \begin{cases} 1 & \text{si } f_d(p_{new}^{ap}) \leq f_d(p_{crt}^{ap}) \\ \exp\left(\frac{f_d(p_{new}^{ap}) - f_d(p_{crt}^{ap})}{T^{ap}}\right) & \text{si } f_d(p_{new}^{ap}) > f_d(p_{crt}^{ap}) \end{cases}$$

Si p_{new}^{ap} est accepté, il devient p_{crt}^{ap} . Si $f_d(p_{crt}^{ap}) < f_d(p_{opt}^{ap})$ alors p_{crt}^{ap} devient p_{opt}^{ap} .

L'agent-processus met alors à jour sa tolérance au changement: si $pa^{ap} = 0$, $pa^{ap} := pa_{init}$ et $T^{ap} := T^{ap} - a$, sinon $pa^{ap} := pa^{ap} - 1$.

Réaction de révision La réaction de révision a pour effet sur l'agent-processus le maintien ou la perte de sa condition de satisfaction de cohérence. Elle est déclenchée par une requête de placement ("Request") en provenance d'un agent-tâche, pour peu que la condition de cohérence soit remplie, mais non la condition de parcours ($cs^{ap} = True$ et $T^{ap} > 0$).

La réaction de révision se déroule de la manière suivante:

1. *Evaluation de l'agent-tâche*: L'agent-processus évalue si la tâche t^{req} de l'agent-tâche requérant a_t^{req} peut remplacer l'une des tâches du processus courant p_{crt}^{ap} . Autrement dit, si t^{req} est une condition OU de G_{AV} pour l'une des tâches de p_{crt}^{ap} .
2. *Mise à jour des connaissances*: Si t^{req} est une alternative d'une tâche de p_{crt}^{ap} , les connaissances de l'agent-processus sont mises à jour de la manière suivante: $cs^{ap} := False$ et p_{new}^{ap} devient le processus partiel constitué de t^{req} et des tâches de p_{crt}^{ap} amont à la tâche remplacée par t^{req} .
3. *Envoi des décisions*: Si $cs^{ap} = False$, l'agent-processus envoie un message d'acceptation à a_t^{req} ("Accepted") et le rajoute à A_t^{ap} . Il envoie également un message de rejet ("Rejected") à tous les agents-tâches $\in A_t^{ap}$ dont la tâche ne fait plus partie de p_{new}^{ap} .

Si $cs^{ap} = True$, l'agent-processus envoie un message de rejet à a_t^{req} ("Rejected")

Réaction de désactivation La réaction de désactivation est déclenchée par la réception d'une requête de placement ("Request"), en provenance d'un agent-tâche arrivant après que les conditions de satisfaction de l'agent-processus soient remplies ($cs^{ap} = True$ et $T^{ap} = 0$).

Dans ce cas, l'agent-processus envoie un message de désactivation ("Deactivate") à l'agent-tâche requérant, ce qui aura pour effet de le bloquer, malgré qu'il soit insatisfait.

Réaction d'activation La réaction d'activation est déclenchée par la réception d'un message, en provenance de l'utilisateur ou de l'agent-flot, requérant un nouveau parcours de l'espace de recherche ("New Search"). Ce message a pour effet la perte de la satisfaction pour l'agent-processus (cohérence et parcours), qui va alors chercher à se satisfaire en activant les agents-tâches.

La réaction d'activation se déroule de la manière suivante:

1. *Mise à jour des connaissances*: à la réception du message, l'agent-processus met à jour ses connaissances: il initialise ses paramètres du recuit simulé ($T^{ap} := T_{init}$ et $pa^{ap} := pa_{init}$)¹, et cs^{ap} devient $cs^{ap} := False$.
2. *Activation des agents-tâches*: l'agent-processus active tous les agents-tâches par la diffusion d'un message d'activation ("Activate") et se met en attente de la première requête.

La réaction d'activation de l'agent-processus constitue le point de départ d'un nouveau parcours de l'espace de recherche.

Réaction d'information La réaction d'information est déclenchée par une demande en provenance de l'utilisateur ou de l'agent-flot ("Process Required"). Elle permet le lien entre le module de sélection et le module de conception.

Selon le contenu du message (*Current* ou *Optimal*), l'agent-processus retourne à l'expéditeur de la demande le contenu de p_{crt}^{ap} ou de p_{opt}^{ap} du moment ("Process").

Dans le cas où p_{crt}^{ap} ou p_{opt}^{ap} n'aurait pas de valeur, l'agent-processus déclenche une réaction d'activation et retourne le premier processus accepté.

Information diffusée

L'information diffusée regroupe les décisions prises par l'agent-processus durant ses réactions. Le tableau A.3 résume les messages envoyés par l'agent-processus.

message	destinataire	contenu	réaction
<i>Accepted</i>	a_t^k	-	intégration révision
<i>Rejected</i>	a_t^k	-	intégration révision
<i>Deactivate</i>	a_t^k	-	désactivation
<i>Activate</i>	a_t^k	-	activation
<i>Process</i>	a_u ou a_i	p_{crt}^{ap} ou p_{opt}^{ap}	information

Tableau A.3: Messages envoyés par l'agent-processus selon ses réactions avec: a_t^k , un agent-tâche; a_u , l'utilisateur; a_i , l'agent-flot; p_{crt}^{ap} , le processus d'assemblage courant du moment; p_{opt}^{ap} , le processus d'assemblage optimal du moment

¹Les valeurs des paramètres du profil de refroidissement du module de sélection sont les suivants: $T_{init} = 40$, elle équivaut à la valeur maximale que peut prendre $f_d(p)$; $pa_{init} = 20$; $a = 5$.

Annexe B

Les agents du module de conception

B.1 Les agents-tâches

B.1.1 Objectifs

Les agents-tâches sont générés par l'agent-flot sur la base du processus opératoire p_{op} sur lequel travaille le module de conception, à raison d'un agent-tâche par tâche du processus. Un agent-tâche a pour objectifs:

- la détermination des ressources génériques, constituant le poste d'assemblage minimal nécessaire à la réalisation de la tâche pour laquelle il a été créé.
- l'intégration de ces ressources à l'ilot d'assemblage dans le cadre d'un poste.

B.1.2 Accointances

Les accointances d'un agent-tâche a_t^j sont:

- a_i , l'agent-flot;
- A_p , l'ensemble des agents-postes.

B.1.3 Mémoire locale

La mémoire locale d'un agent-tâche contient ses connaissances. Les connaissances de l'agent-tâche a_t^j sont décrites par l'heptuplet:

$C_{a_t^j} = \langle t^j, N0^j, N1^j, OP^j, SP^j, a_p^j, A_{pr}^j \rangle$ avec:

- t^j , la tâche que doit réaliser l'agent-tâche. Cette tâche peut être une tâche constitutive (géométrique, physique ou complémentaire), ou une tâche positionnelle (réorientation, chargement, déchargement);

- NO^j , l'ensemble des ressources génériques du niveau NO, constituant le poste minimal. Le niveau NO d'un poste est le lieu de la transformation effective des constituants¹;
- $N1^j$, l'ensemble des ressources génériques du niveau N1, constituant le poste minimal. Le niveau N1 est le lieu de distribution d'un constituant isolé, positionné et orienté;
- OP^j , l'ensemble des ressources génériques constituant les effecteurs du poste minimal;
- SP^j , l'ensemble des ressources génériques constituant les équipements spécialisés du poste minimal;
- a_p^j l'agent-poste $\in A_p$ qui a effectivement intégré les ressources génériques de a_t^j .
- A_{pr}^j , le sous-ensemble de A_p des agents-postes qui ont définitivement rejeté les requêtes de a_t^j .

B.1.4 Comportement

Le comportement d'un agent-tâche est de type réactif et il est guidé par la recherche de satisfaction: la réception d'un message en provenance d'une acointance déclenche une réaction appropriée, afin de se rapprocher des conditions de satisfaction.

Conditions de satisfaction

- *Condition de satisfaction technologique*: l'ensemble des ressources générique a été généré ($NO^j, N1^j, OP^j, SP^j \neq \{\emptyset\}$).
- *Condition de satisfaction d'intégration*: l'ensemble des ressources génériques est intégré dans un poste po_{crt}^k d'un agent-poste a_p^k ($a_p^j \neq \{\emptyset\}$).

Condition d'existence

La condition d'existence d'un agent-tâche est que la tâche t^j pour laquelle l'agent a été créé fasse partie du processus opératoire p_{op} sur lequel travaille le SMA de conception.

Cette condition n'est pas vérifiée par l'agent-tâche lui-même, mais par l'agent-flot lors des changements de processus opératoire.

Réactions

La réaction déclenchée par l'agent-tâche dépend du message reçu. Le tableau B.1 donne les conditions de déclenchement des réactions d'un agent-tâche.

Les réactions d'initialisation et de suicide sont liées à la condition d'existence de l'agent-tâche. Les réactions de rejet, de refus et d'acceptation sont liées à la condition d'intégration.

¹Cf. §5.4.1

message	expéditeur	contenu	réaction
<i>Activate</i>	a_i	-	initialisation
<i>Suicide</i>	a_i	-	suicide
<i>Rejected</i>	a_p^k	-	rejet
<i>Refused</i>	a_p^k	a_p^k	refus
<i>Accepted</i>	a_p^k	a_p^k	acceptation

Tableau B.1: Conditions de déclenchement des réactions d'un agent-tâche avec: a_i , l'agent-flot; a_p^k , un agent-poste

Réaction d'initialisation La réaction d'initialisation constitue la première réaction de la vie de l'agent-tâche. Elle lui permet de remplir sa condition de satisfaction technologique par la génération des ressources génériques et de tenter de remplir sa condition d'intégration par l'envoi d'une requête à un agent-poste.

La réaction d'initialisation est déclenchée par la réception d'un message d'activation en provenance de l'agent-flot ("Activate"). Elle se déroule de la manière suivante:

1. *Initialisation des connaissances*: les connaissances de l'agent-tâche a_i^j sont initialisées avec les valeurs suivantes:
 - $t^j = t^j$, la tâche pour laquelle l'agent-tâche a été créé;
 - $NO^j = N1^j = OP^j = SP^j = a_p^j = A_{pr}^j = \{\emptyset\}$
2. *Création des ressources génériques*: sur la base de la tâche t^j et des constituants impliqués, l'agent-tâche crée ses ressources génériques².
Le résultat de cette étape est stocké dans NO^j , $N1^j$, OP^j et SP^j .
3. *Choix ou création d'un nouvel agent-poste*: si l'agent-tâche n'a pas d'alternative pour se placer ($A_{pr}^j = A_p$), alors il crée un nouvel agent-poste a_p^{new} auquel il enverra sa requête, et la population des agents-postes est modifiée comme suit: $A_p = A_p \cup \{a_p^{new}\}$. Sinon il choisit aléatoirement un agent-poste dans $A_p \setminus A_{pr}^j$.
4. *Envoi d'une requête*: l'agent-tâche envoie une requête de placement à l'agent-poste retenu ("Request").

Réaction de suicide La réaction de suicide constitue la dernière réaction de la vie de l'agent-tâche: sa condition d'existence n'est plus remplie et il doit disparaître. Ce cas peut survenir lors des changements de processus opératoire.

La réaction de suicide est déclenchée par la réception d'un ordre de suicide en provenance de l'agent-flot ("Suicide"). Elle se déroule de la manière suivante:

Suicide: l'agent-tâche, si il est placé sur un agent-poste, envoie un avis de décès ("Dead") à a_p^j et la population des agents-tâches devient alors $A_i^{pop} = A_i^{pop} \setminus \{a_i^j\}$. L'agent-tâche a cessé d'exister.

²Pour plus de détails, voir l'annexe D.1

Réaction de rejet La réaction de rejet est déclenchée par la réception d'un message de rejet de la part d'un agent-poste α_p^k ("Rejected"). Ce message a pour effet sur l'agent-tâche, la perte de la satisfaction d'intégration. En réaction, il envoie une nouvelle requête à un agent-poste.

La réaction de rejet se déroule de la manière suivante:

1. *Modification des connaissances*: à la réception du message de rejet, les connaissances de l'agent-tâche sont modifiées de la manière suivante: $\alpha_p^j = \{\emptyset\}$.
2. *Choix d'un agent-poste*: le choix de l'agent-poste destinataire du message se fait de manière aléatoire, tout agent-poste $\in (A_p \setminus A_{pr}^j)$ étant potentiellement candidat pour intégrer l'agent-tâche.
3. *Envoi d'une requête*: l'agent-tâche envoie à l'agent-poste retenu une requête de placement ("Request").

Réaction de refus La réaction de refus est déclenchée par le refus définitif de l'agent-poste α_p^k à traiter une requête de l'agent-tâche α_t^j . Ce message a pour effet sur l'agent-tâche, la perte de la satisfaction d'intégration. En réaction, il envoie une nouvelle requête à un agent-poste.

La réaction de refus se déroule de la manière suivante:

1. *Modification des connaissances*: à la réception du message de refus, les connaissances de l'agent-tâche sont modifiées de la manière suivante: $\alpha_p^j = \{\emptyset\}$ et $A_{pr}^j = A_{pr}^j \cup \{\alpha_p^k\}$.
2. *Choix ou création d'un nouvel agent-poste*: si l'agent-tâche n'a pas d'alternative pour se placer ($A_{pr}^j = A_p$), alors il crée un nouvel agent-poste α_p^{new} auquel il enverra sa requête et la population des agents-postes est modifiée comme suit: $A_p = A_p \cup \{\alpha_p^{new}\}$. Sinon il choisit aléatoirement un agent-poste dans $A_p \setminus A_{pr}^j$.
3. *Envoi d'une requête*: l'agent-tâche envoie une requête de placement à l'agent-poste retenu ("Request").

Réaction d'acceptation La réaction d'acceptation est déclenchée par la réception d'un message d'acceptation en provenance d'un agent-poste α_p^k ("Accepted"). Ce message a pour effet sur l'agent-tâche, la satisfaction de la condition d'intégration. Elle ne fait que modifier les connaissances de l'agent-tâche de la manière suivante: $\alpha_p^j = \alpha_p^k$.

Information diffusée

Le tableau B.2 résume les messages envoyés par un agent-tâche.

B.2 Les agents-postes

B.2.1 Objectifs

Les agents-postes sont générés par les agents-tâches de manière dynamique. Le premier agent-poste est créé par le premier agent-tâche activé. Les agents-postes

message	destinataire	contenu	réaction
<i>Request</i>	a_p^k	a_t^j	initialisation refus rejet
<i>Dead</i>	a_p^j	a_t^j	suicide

Tableau B.2: Messages envoyés par un agent-tâche selon ses réactions avec: a_p^k , un agent-poste; a_t^j , l'agent-tâche; a_p^j , l'agent-poste sur lequel est placé l'agent-tâche

suivants sont créés par les agents-tâches refusés qui n'ont plus d'alternative pour se placer.

Les objectifs d'un agent-poste sont les suivants:

- la création et l'organisation des équipements qui vont constituer un poste d'assemblage, sur la base des ressources génériques des agents-tâches, de manière à respecter les contraintes de précédence et le temps de cycle contrat.
- l'intégration du poste à l'ilot d'assemblage dans le cadre d'une cellule.

B.2.2 Accointances

Les accointances d'un agent-poste a_p^j sont constituées de:

- a_i , l'agent-slot;
- A_c , l'ensemble des agents-cellules;
- $A_t^{p\circ p}$, l'ensemble des agents-tâches du module de conception.

B.2.3 Mémoire locale

La mémoire locale d'un agent-poste contient ses connaissances. Les connaissances d'un agent-poste a_p^j sont décrites par l'octuplet:

$$C_{a_p^j} = \langle p_{op}, T_{cc}, t_t^j, po_{crt}^j, po_{new}^j, A_{cr}^j, a_c^j, A_t^j, cs_{tech}^j \rangle \text{ avec:}$$

- p_{op} , le processus opératoire sur lequel travaille le SMA de conception;
- T_{cc} , le temps de cycle contrat;
- t_t^j , le temps de transfert entre les postes de la cellule qui a intégré le poste de a_p^j .
- po_{crt}^j , le poste d'assemblage courant de a_p^j . Un poste d'assemblage est décrit par le sextuplet:

$$po^k = \langle t_{cp}^k, p_p^k, E_{N0}^k, E_{N1}^k, E_{OP}^k, E_{SP}^k \rangle \text{ avec:}$$

- t_{cp}^k , le temps de cycle du poste;
- p_p^k , la partie du processus p_{op} exécutée dans le poste;

- E_{N0}^k , l'ensemble des équipements du niveau N0 du poste;
 - E_{N1}^k , l'ensemble des équipements du niveau N1 du poste;
 - E_{OP}^k , l'ensemble des équipements qui sont les opérateurs du poste;
 - E_{SP}^k , l'ensemble des équipements spécialisés du poste.
- po_{new}^j , le nouveau poste d'assemblage créé par a_p^j , et qui pourra constituer le nouvel état de po_{crt}^j .
 - A_{cr}^j , l'ensemble des agents-cellules $\in A_c$ qui ont définitivement rejeté les requêtes de a_p^j ;
 - a_c^j , l'agent-cellule qui a effectivement intégré le poste de a_p^j ;
 - A_t^j , l'ensemble des agents-tâches $\in A_t^{pp}$ dont les ressources génériques ont été intégrées par a_p^j dans son poste courant po_{crt}^j .
 - cs_{tech}^j , valeur booléenne indiquant si la condition de satisfaction technologique est remplie ou non

B.2.4 Comportement

Le comportement d'un agent-poste est de type réactif et il est guidé par la recherche de satisfaction: la réception d'un message en provenance d'une acointance déclenche une réaction appropriée, afin de se rapprocher des conditions de satisfaction.

Condition de satisfaction

Les conditions de satisfaction d'un agent-poste sont:

- *Condition de satisfaction technologique*: la valeur de $|T_{cc} - t_{cp}^{crt}|$, la différence entre le temps de cycle contrat et le temps de cycle de po_{crt}^j , est minimale ($cs_{tech}^j = True$).
- *Condition de satisfaction d'intégration*: le poste d'assemblage qu'il a créé po_{crt}^j est intégré dans une cellule ($a_c^j \neq \{\emptyset\}$).

Condition d'existence

La condition d'existence d'un agent-poste est qu'il y ait toujours au moins un agent-tâche placé dans l'agent-poste: $A_t^j \neq \{\emptyset\}$.

Cette condition est testée par l'agent-tâche en cas de modification de A_t^j à la suite du rejet d'agents-tâches déjà placés. Ceci peut se produire lors de la réactivation de l'agent-poste ou lorsque des agents-tâches disparaissent après un changement de processus opératoire.

message	expéditeur	contenu	cs_{tech}^j	réaction
<i>Request</i>	a_t^k	a_t^k	<i>False</i> <i>True</i>	intégration satisfaction
<i>Rejected</i>	a_c^k	-	<i>True</i> ou <i>False</i>	rejet
<i>Refused</i>	a_c^k	a_c^k	<i>True</i> ou <i>False</i>	refus
<i>Accepted</i>	a_c^k	a_c^k	<i>True</i> ou <i>False</i>	acceptation
<i>Dead</i>	a_t^k	a_t^k	<i>True</i> ou <i>False</i>	révision
<i>Reactivate</i>	a_i	T^{a_i}/T_{init}	<i>True</i> ou <i>False</i>	réactivation

Tableau B.3: Conditions de déclenchement des réactions d'un agent-poste avec: a_t^k , un agent-tâche; a_c^k , un agent-cellule; a_i , l'agent-flot

Réactions

La réaction déclenchée par l'agent-poste dépend du message reçu et de l'état de satisfaction technologique de l'agent. Le tableau B.3 donne les conditions de déclenchement des réactions d'un agent-poste.

Les réactions d'intégration et de satisfaction sont liées à la condition technologique de l'agent-poste. Les réactions de rejet, de refus et d'acceptation sont liées à la condition d'intégration. Les réactions de révision et de réactivation sont liées aux conditions d'existence et de satisfaction technologique de l'agent-poste.

Réaction d'intégration La réaction d'intégration permet à l'agent-poste de se rapprocher de sa condition de satisfaction technologique, et de tenter de remplir sa condition d'intégration par l'envoi d'une requête à un agent-cellule.

La réaction d'intégration est déclenchée par une requête de placement en provenance d'un agent-tâche a_t^{req} ("Request"), pour peu que la condition technologique ne soit pas encore remplie ($cs_{tech}^j = False$).

La réaction d'intégration se déroule de la manière suivante:

1. *Evaluation de l'agent-tâche*: cette évaluation permet de satisfaire les contraintes de précédence. Selon la position, dans le processus p_{op} , de la tâche t^{req} de l'agent-tâche requérant a_t^{req} par rapport à p_p^{crt} , la partie du processus exécutée dans po_{crt}^j , l'agent-poste exécute les actions suivantes:
 - Si t^{req} est située en amont de p_p^{crt} : l'agent-poste envoie un message de refus ("Refused") à l'agent-tâche requérant.
 - Si t^{req} est située en aval de p_p^{crt} : l'agent-poste envoie un message de rejet ("Rejected") à l'agent-tâche requérant.
 - Si t^{req} est située *directement* en aval de p_p^{crt} ou que $po_{crt}^j = \{\emptyset\}$: l'agent-poste crée un nouveau poste.
2. *Création d'un nouveau poste*: si l'agent-tâche n'a pas été rejeté après son évaluation, l'agent-poste génère un nouveau poste po_{new}^j sur la base de po_{crt}^j et des ressources génériques de a_t^{req} ³. Le poste ainsi créé est stocké dans po_{new}^j .

³Pour plus de détails, voir l'annexe D.2

Si $|T_{cc} - t_{cp}^{crt}| > |T_{cc} - t_{cp}^{new}|$, le nouveau poste po_{new}^j est accepté et devient le poste courant po_{crt}^j . Sinon l'agent-poste crée un nouveau poste d'assemblage, sur la base de po_{new}^j sans tenir compte des ressources génériques de l'agent-tâche *amont* de p_p^{new} . Si après cette deuxième itération, po_{new}^j n'est toujours pas accepté, c'est que $|T_{cc} - t_{cp}^{crt}|$ est minimale et la condition de satisfaction est remplie: $cs_{tech}^j = True$

3. Envoi des décisions:

- En ce qui concerne a_t^{req} , l'agent-tâche requérant: si po_{new}^j a été accepté, l'agent-poste envoie à a_t^{req} un message d'acceptation ("Accepted") et l'ajoute dans A_t^j , sinon il lui envoie un message de refus ("Refused").
- L'agent-poste envoie, si po_{new}^j a été accepté à la deuxième itération, un message de refus ("Refused") à l'agent-tâche amont dont les ressources génériques ont été écartées durant l'élaboration du nouveau poste et le retire de A_t^j .
- Si po_{new}^j a été accepté, l'agent-poste envoie un message de modification ("Modified") à l'agent-cellule sur lequel il est placé.

4. Contrôle de la condition d'intégration: si $A_c^j = \{\emptyset\}$, l'agent-poste va chercher à se placer en envoyant une requête à un agent-cellule:

- Choix ou création d'un nouvel agent-cellule: si l'agent-poste n'a pas d'alternative pour se placer ($A_{cr}^j = A_c$) alors il crée un nouvel agent-cellule a_c^{new} auquel il enverra sa requête et la population des agents-cellules devient: $A_c = A_c \cup \{a_c^{new}\}$. Sinon il choisit aléatoirement un agent-cellule dans $A_c \setminus A_{cr}^j$.
- Envoi d'une requête: l'agent-poste envoie une requête de placement à l'agent-cellule retenu ("Request").

Réaction de satisfaction La réaction de satisfaction permet à l'agent-tâche de maintenir sa condition de satisfaction technologique, et de tenter de remplir sa condition d'intégration par l'envoi d'une requête à un agent-cellule.

La réaction de satisfaction est déclenchée par une requête de placement en provenance d'un agent-tâche a_t^{req} ("Request") alors que la condition technologique est déjà remplie ($cs_{tech}^j = True$).

La réaction de satisfaction se déroule de la manière suivante:

1. *Rejet de l'agent-tâche*: l'agent-poste envoie un message de refus à a_t^{req} ("Refused").
2. *Contrôle de la condition d'intégration*: si $A_c^j = \{\emptyset\}$, l'agent-poste va chercher à se placer en envoyant une requête à un agent-cellule:
 - Choix ou création d'un nouvel agent-cellule: si l'agent-poste n'a pas d'alternative pour se placer ($A_{cr}^j = A_c$), alors il crée un nouvel agent-cellule a_c^{new} auquel il enverra sa requête, et la population des agents-cellules devient: $A_c = A_c \cup \{a_c^{new}\}$. Sinon il choisit aléatoirement un agent-cellule dans $A_c \setminus A_{cr}^j$.

- Envoi d'une requête: l'agent-poste envoie une requête de placement à l'agent-cellule retenu ("Request").

Réaction de rejet La réaction de rejet est déclenchée par la réception d'un message de rejet de la part d'un agent-cellule a_c^k ("Rejected"). Ce message a pour effet sur l'agent-poste, la perte de la satisfaction d'intégration. En réaction, il envoie une nouvelle requête à un agent-cellule.

La réaction de rejet se déroule de la manière suivante:

1. *Modification des connaissances*: à la réception du message de rejet, les connaissances de l'agent-poste sont modifiées de la manière suivante: $a_c^j = \{\emptyset\}$.
2. *Choix d'un agent-cellule*: le choix de l'agent-cellule destinataire du message se fait de manière aléatoire, tout agent-cellule $\in (A_c \setminus A_{cr}^j)$ étant potentiellement candidat pour intégrer l'agent-poste.
3. *Envoi d'une requête*: l'agent-poste envoie à l'agent-cellule retenu une requête de placement ("Request").

Réaction de refus La réaction de refus est déclenchée par le refus définitif de l'agent-cellule a_c^k à traiter une requête de l'agent-poste a_p^j . Ce message a pour effet sur l'agent-poste, la perte de la satisfaction d'intégration. En réaction, il envoie une nouvelle requête à un agent-cellule.

La réaction de refus se déroule de la manière suivante:

1. *Modification des connaissances*: à la réception du message de refus, les connaissances de l'agent-poste sont modifiées de la manière suivante: $a_c^j = \{\emptyset\}$ et $A_{cr}^j = A_{cr}^j \cup \{a_c^k\}$.
2. *Choix ou création d'un nouvel agent-cellule*: si l'agent-poste n'a pas d'alternative pour se placer ($A_{cr}^j = A_c$), alors il crée un nouvel agent-cellule a_c^{new} auquel il enverra sa requête, et il modifie ses accointances: $A_c = A_c \cup \{a_c^{new}\}$. Sinon il choisit aléatoirement un agent-cellule dans $A_c \setminus A_{cr}^j$.
3. *Envoi d'une requête*: l'agent-poste envoie une requête de placement à l'agent-cellule retenu ("Request").

Réaction d'acceptation Cette réaction est déclenchée par la réception d'un message d'acceptation en provenance d'un agent-cellule a_c^k ("Accepted"). Ce message a pour effet sur l'agent-poste, la satisfaction de la condition d'intégration. Elle ne fait que modifier les connaissances de l'agent-poste de la manière suivante: $a_c^j = a_c^k$.

Réaction de révision La réaction de révision est déclenchée par la réception d'un avis de décès en provenance d'un agent-tâche ("Dead"). Ce message a pour effet sur l'agent-poste, la modification de son poste, avec pour conséquence la perte de la satisfaction technologique et la remise en cause éventuelle de la condition d'existence.

La réaction de révision se déroule de la manière suivante:

1. *Mise à jour des connaissances*: l'agent-poste, afin de tenir compte de la disparition de l'agent-tâche suicidé a_t^j , met à jour ses connaissances:
 - $A_t^j = A_t^j \setminus \{a_t^j\}$
 - $cs_{tech}^j = False$
2. *Contrôle de la condition d'existence*: si $A_c^j = \{\emptyset\}$ alors l'agent-poste se suicide: si il est placé sur un agent-cellule, il envoie un avis de décès ("Dead") à a_c^j et la population d'agents-postes devient $A_p = A_p \setminus \{a_p^j\}$. La vie de l'agent-poste s'arrête alors ici.
3. *Création d'un nouveau poste*: la génération du nouveau poste se fait sur l'ensemble des ressources génériques des agents-tâches $\in A_t^j$ qui couvrent la plus grande partie de p_{op} . Cette étape se déroule comme l'étape 2. de la réaction d'intégration, si ce n'est que le poste obtenu po_{new}^j est automatiquement accepté comme le poste courant.
4. *Envoi des décisions*:
 - l'agent-poste envoie à tous les agents-tâches qui ne font pas partie du nouveau poste courant po_{crt}^j un message de rejet ("Rejected"), et les retire de A_t^j .
 - l'agent-poste envoie un message de modification ("Modified") à l'agent-cellule sur lequel il est placé.
5. *Contrôle de la condition d'intégration*: si $A_c^j = \{\emptyset\}$, l'agent-poste va chercher à se placer en envoyant une requête à un agent-cellule:
 - Choix ou création d'un nouvel agent-cellule: si l'agent-poste n'a pas d'alternative pour se placer ($A_{cr}^j = A_c$), alors il crée un nouvel agent-cellule a_c^{new} auquel il enverra sa requête, et la population des agents-cellules devient: $A_c = A_c \cup \{a_c^{new}\}$. Sinon il choisit aléatoirement un agent-cellule dans $A_c \setminus A_{cr}^j$.
 - Envoi d'une requête: l'agent-poste envoie une requête de placement à l'agent-cellule retenu ("Request").

Réaction de réactivation La réaction de réactivation est déclenchée à la réception d'un message de réactivation en provenance de l'agent-flot ("Reactivate"). Comme dans la réaction de révision, l'agent-poste doit modifier son poste avec pour conséquence la perte de la satisfaction technologique et la remise en cause éventuelle de sa condition d'existence.

La réaction de réactivation se déroule comme la réaction de révision, si ce n'est que la première étape (mise à jour des connaissances) est remplacée par une étape de rejet d'agents-tâches.

Rejet d'agents-tâches: l'agent-poste rejette les agents-tâches $\in A_t^j$ avec une probabilité de rejet égale à T^{ai}/T_{init} fournie dans le message. Il envoie un message de rejet ("Rejected") aux agents-tâches concernés et les retire de A_t^j . Si il y a eu rejet d'agent-tâche, l'agent-poste modifie aussi ses connaissances: $cs_{tech}^j = False$.

La réaction de réactivation se déroule alors comme la réaction de révision: contrôle de la condition d'existence; génération d'un nouveau poste po_{new}^j sur la base de la partie de p_{op} la plus grande que couvrent les agents-tâches $\in A_t^j$; envoi des décisions prises par l'agent-poste; contrôle de la condition d'intégration.

Information diffusée

L'information diffusée regroupe les décisions prises par l'agent-poste durant ses réactions. Le tableau B.4 résume les messages envoyés par un agent-poste.

message	destinataire	contenu	réaction
<i>Accepted</i>	a_t^k	a_p^j	intégration
<i>Rejected</i>	a_t^k	-	intégration révision réactivation
<i>Refused</i>	a_t^k	a_p^j	intégration satisfaction
<i>Request</i>	a_c^k	a_p^j	intégration satisfaction rejet refus révision réactivation
<i>Modified</i>	a_c^j	a_p^j	intégration révision réactivation
<i>Dead</i>	a_c^j	a_p^j	révision réactivation

Tableau B.4: Messages envoyés par un agent-poste selon ses réactions avec: a_t^k , un agent-tâche; a_p^j , l'agent-poste; a_c^k , un agent-cellule; a_c^j , l'agent-cellule sur lequel est placé l'agent-poste

B.3 Les agents-cellules

B.3.1 Objectifs

Les agents-cellules sont générés par les agents-postes de manière dynamique. Le premier agent-cellule est créé par le premier agent-poste activé. Les agents-cellules suivants sont créés par les agents-postes refusés qui n'ont plus d'alternative pour se placer.

Les objectifs d'un agent-cellule sont:

- la création des équipements de transfert entre les postes, fournis par les agents-postes, qui vont constituer une cellule d'assemblage, de manière à respecter les contraintes de précedence et les contraintes-ressources.
- l'intégration de la cellule à l'flot d'assemblage.

B.3.2 Accointances

Les accointances d'un agent-cellule a_c^j sont constituées de:

- a_i , l'agent-flot;
- A_p , l'ensemble des agents-postes.

B.3.3 Mémoire locale

La mémoire locale d'un agent-cellule contient ses connaissances. Les connaissances d'un agent-cellule a_c^j sont décrites par le quintuplet:

$C_{a_c^j} = \langle p_{op}, c_{crt}^j, c_{new}^j, A_p^j, cs_{tech}^j \rangle$ avec:

- p_{op} , le processus d'assemblage sur lequel travaille le SMA de conception.
- c_{crt}^j , la cellule d'assemblage courante de a_c^j . Une cellule d'assemblage c^k est décrite par le quadruplet:
 $c^k = \langle t_{cc}^k, p_c^k, P_k^i, T_k^i \rangle$ avec:
 - t_{cc}^k , le temps de cycle de la cellule
 - p_c^k , la partie du processus p_{op} exécutée dans la cellule.
 - P_k^i , l'ensemble des postes de la cellule
 - T_k^i , l'ensemble des systèmes de transfert entre les postes de la cellule
- c_{new}^j , la nouvelle cellule d'assemblage créée par a_c^j et qui pourra constituer la nouvelle valeur de c_{crt}^j .
- A_p^j , l'ensemble des agents-postes intégrés par a_c^j
- cs_{tech}^j , paramètre caractérisant la condition de satisfaction technologique: il peut prendre les valeurs "True" si la condition est remplie en amont et en aval de la cellule c_{crt}^j , "Amont", si elle est remplie en amont de la cellule, "Aval" si c'est en aval, et "False" si la condition n'est pas remplie.

B.3.4 Comportement

Le comportement d'un agent-cellule est de type réactif et il est guidé par la recherche de satisfaction: la réception d'un message en provenance d'une accointance déclenche une réaction appropriée, afin de se rapprocher des conditions de satisfaction.

Condition de satisfaction

Les conditions de satisfaction d'un agent-cellule sont:

- *Condition de satisfaction technologique*: la cellule créée doit constituer l'ensemble cohérent de postes le plus grand possible ($cs_{tech}^j = True$).
- *Condition de satisfaction d'intégration*: la cellule d'assemblage qu'il a créée c_{crt}^j est intégrée dans l'îlot d'assemblage. Cette condition est remplie, l'agent-ilot acceptant automatiquement un agent-cellule, dès le moment où l'agent-cellule a signalé son existence par l'envoi d'un message à l'agent-ilot.

Condition d'existence

La condition d'existence d'un agent-cellule est qu'il y ait toujours au moins un agent-poste placé dans l'agent-cellule: $A_p^j \neq \{\emptyset\}$.

Cette condition est testée par l'agent-cellule à chaque modification de A_p^j à la suite du rejet d'agents-postes déjà placés. Ceci peut se produire à chaque révision de la cellule lorsque des agents-postes ont modifié leur poste ou ont disparu.

Réactions

La réaction déclenchée par l'agent-cellule dépend du message reçu et de l'état de satisfaction technologique de l'agent. Le tableau B.5 donne les conditions de déclenchement des réactions d'un agent-cellule.

message	expéditeur	contenu	cs_{tech}^j	réaction
<i>Request</i>	a_p^k	a_p^k	$cs_{tech}^j \neq True$ $cs_{tech}^j = True$	intégration satisfaction
<i>Dead</i>	a_p^k	a_p^k	indép. de cs_{tech}^j	révision
<i>Modified</i>	a_p^k	a_p^k	indép. de cs_{tech}^j	révision

Tableau B.5: Conditions de déclenchement des réactions d'un agent-cellule avec: a_p^k , un agent-poste

Les réactions d'intégration et de satisfaction sont liées à la condition technologique de l'agent-cellule. La réaction de révision est liée aux conditions d'existence et de satisfaction technologique de l'agent-cellule.

Réaction d'intégration La réaction d'intégration permet à l'agent-cellule de se rapprocher de sa condition de satisfaction technologique. Elle est déclenchée par la requête de placement en provenance d'un agent-poste a_p^{req} ("Request"), pour peu que la condition de satisfaction technologique ne soit pas entièrement remplie $cs_{tech}^j \neq True$.

La réaction d'intégration se déroule de la manière suivante:

1. *Evaluation de l'agent-poste*: cette évaluation permet de satisfaire les contraintes de précédence. Selon la position dans le processus p_{op} de p_p^{req} , la partie du processus exécutée dans le poste de l'agent-poste requérant, par rapport à p_c^j , l'agent-cellule exécute les actions suivantes:
 - Si p_p^{req} est située en amont de p_c^j : l'agent-cellule envoie un message de rejet ("Rejected") à l'agent-poste requérant.
 - Si p_p^{req} est située directement en amont de p_c^j et que $cs_{tech}^j \neq Amont$, l'agent-cellule détermine la nouvelle cellule. Sinon il envoie un message de refus à a_p^{req} ("Refused").
 - Si p_p^{req} est située directement en aval de p_c^j et que $cs_{tech}^j \neq Aval$, l'agent-cellule détermine la nouvelle cellule. Sinon il envoie un message de refus à a_p^{req} ("Refused").

- Si p_p^{req} est située en aval de p_c^j : l'agent-cellule envoie un message de rejet ("Rejected") à l'agent-poste requérant.
2. *Création d'une nouvelle cellule*: si l'agent-poste n'a pas été rejeté après son évaluation, l'agent-cellule génère une nouvelle cellule c_{new}^j sur la base de c_{crt}^j et du poste de a_p^{req} . La première étape consiste à déterminer si le poste de a_p^{req} peut être intégré à la nouvelle cellule. Ceci est déterminé de la manière suivante:
- si le poste de a_p^{req} possède un équipement NO, en entrée ou en sortie, (selon que a_p^{req} soit aval ou amont à p_c^j) qui peut recevoir le sous-assemblage commun (composition et orientation) à tous les postes, il peut être intégré à la nouvelle cellule.
 - si c'est le cas et qu'il y a plusieurs postes, l'agent-cellule choisit le système de transfert qui, selon la définition de la cellule, est un transfert rigide.
 - si le nombre de postes de la nouvelle cellule ainsi obtenue ne dépasse pas le maximum fixé par le type de système de transfert (pour des raisons de fiabilité), le poste de a_p^{req} est définitivement accepté.

La seconde étape consiste dans la mise à jour de la condition technologique:

- Si a_p^{req} a été accepté:
 - c_{new}^j devient c_{crt}^j . La description de la cellule est complétée par son temps de cycle: $t_{cc}^{crt} = \max(t_{cp}^1, \dots, t_{cp}^i)$, le temps de cycle du poste le plus lent.
 - cs_{tech}^j est mis à jour: Si le poste de a_p^{req} possède des équipements NO distincts en entrée et en sortie:
 - * si a_p^{req} est aval à p_c^j , $cs_{tech}^j = Aval$ (ou égale à "True" si cs_{tech}^j était égale à "Amont")
 - * si a_p^{req} est amont à p_c^j , $cs_{tech}^j = Amont$ (ou égale à "True" si cs_{tech}^j était égale à "Aval")
 Sinon, cs_{tech}^j ne change pas.
 - Si a_p^{req} a été refusé: cs_{tech}^j est mise à jour de la manière suivante:
 - si a_p^{req} est aval à p_c^j , $cs_{tech}^j = Aval$ (ou égale à "True" si cs_{tech}^j était égale à "Amont")
 - si a_p^{req} est amont à p_c^j , $cs_{tech}^j = Amont$ (ou égale à "True" si cs_{tech}^j était égale à "Aval")
3. *Envoi des décisions*:
- Si a_p^{req} a été intégré, l'agent-cellule lui envoie un message d'acceptation ("Accepted"), sinon il lui envoie un message de refus a_p^{req} ("Refused").
 - L'agent-cellule envoie un message de modification ("Modified") à l'agent-flot.

Réaction de satisfaction La réaction de satisfaction permet à l'agent-cellule de maintenir sa condition de satisfaction technologique. Elle est déclenchée par la requête de placement en provenance d'un agent-poste a_p^{req} ("Request"), alors que la condition technologique est entièrement remplie $cs_{tech}^j = True$.

La réaction de satisfaction consiste simplement dans l'envoi d'un message de refus à a_p^{req} ("Refused").

Réaction de révision La réaction de révision est déclenchée par la réception d'un avis de décès ("Dead") ou de modification ("Modified") en provenance d'un agent-poste a_p^k placé dans l'agent-cellule. L'agent-cellule doit modifier sa cellule avec pour conséquence la perte de la satisfaction technologique, et la remise en cause éventuelle de sa condition d'existence.

La réaction de révision se déroule de la manière suivante:

1. *Mise à jour des connaissances*: l'agent-cellule met à jour ses connaissances:
 - si a_p^k s'est suicidé: $A_p^j = A_p^j \setminus \{a_p^k\}$
 - si a_p^k s'est modifié: $A_p^j = A_p^j \setminus \{a_p^k\}$ et l'agent-cellule envoie un message de rejet ("Rejected") à a_p^k .
2. *Contrôle des conditions d'existence*: si, après la mise à jour, $A_p^j = \{\emptyset\}$ alors l'agent-cellule se suicide: il envoie un avis de décès ("Dead") à l'agent-flot, et la population d'agents-cellules devient $A_c = A_c \setminus \{a_c^j\}$. Il a fini d'exister.
3. *Création d'une nouvelle cellule*: la génération de la nouvelle cellule se fait sur l'ensemble des postes des agents-postes $\in A_p^j$ qui couvrent la plus grande partie de p_{op} . Si cette partie est couverte par plusieurs postes, l'agent-cellule choisit un nouveau système de transfert. La nouvelle cellule c_{new}^j ainsi générée devient alors automatiquement c_{crt}^j .

La condition technologique est alors mise à jour de la manière suivante:

- si a_p^k se trouvait en aval de p_c^{crt} , la partie de processus maintenant couverte par c_{crt}^j : $cs_{tech}^j = False$ si cs_{tech}^j était égale à "Aval" (ou égale à "Amont" si cs_{tech}^j était égale à "True"). Sinon elle reste égale à "False".
 - si a_p^k se trouvait en amont de p_c^{crt} , la partie de processus maintenant couverte par c_{crt}^j : $cs_{tech}^j = False$ si cs_{tech}^j était égale à "Amont" (ou égale à "Aval" si cs_{tech}^j était égale à "True"). Sinon elle reste égale à "False".
4. *Envoi des décisions*:
 - l'agent-cellule envoie à tous les agents-postes qui ne font pas partie de la nouvelle cellule courante c_{crt}^j un message de rejet ("Rejected") et les retire de A_p^j .
 - l'agent-cellule envoie un message de modification ("Modified") à l'agent-flot.

Information diffusée

L'information diffusée regroupe les décisions prises par l'agent-cellule durant ses réactions. Le tableau B.6 résume les messages envoyés par un agent-cellule.

message	destinataire	contenu	réaction
<i>Accepted</i>	a_p^k	a_c^j	intégration
<i>Refused</i>	a_p^k	a_c^j	intégration satisfaction
<i>Modified</i>	a_i	a_c^j	intégration révision
<i>Rejected</i>	a_p^k	-	révision
<i>Dead</i>	a_i	a_c^j	révision

Tableau B.6: Messages envoyés par un agent-cellule selon ses réactions avec: a_p^k , un agent-poste; a_c^j , l'agent-cellule; a_i , l'agent-flot

B.4 L'agent-flot

B.4.1 Objectifs

Il n'y a qu'un seul agent-flot généré par le système. L'agent-flot a pour objectifs:

- la création des équipements de transfert et l'organisation des cellules qui vont constituer l'îlot d'assemblage, sur la base des cellules des agents-cellules, de manière à respecter les contraintes de précédence et les contraintes-ressources
- le contrôle du parcours de l'espace physique, par la gestion de la population des agents-tâches et du recuit simulé sur le critère du coût technologique ($f_c(i) = \sum_{j=1}^{n_c} (\sum_{k=1}^{n_p^j} (\sum_{m=1}^{n_e^k} c_e^m) + c_{et}^j) + c_{et}^i$).

B.4.2 Accointances

Les accointances de l'agent-flot sont:

- A_t^{op} , l'ensemble des agents-tâches du module de conception;
- A_p , l'ensemble des agents-postes;
- A_c , l'ensemble des agents-cellules;
- a_u , l'utilisateur du système;
- a_p , l'agent-processus.

B.4.3 Mémoire locale

La mémoire locale de l'agent-flot contient ses connaissances. Les connaissances de l'agent-flot a_i sont décrites par l'heptuplet:

$C_{a_i} = \langle p_{op}, T^{a_i}, pa^{a_i}, i_{crt}^{a_i}, i_{new}^{a_i}, i_{opt}^{a_i}, cs_{tech}^{a_i} \rangle$ avec:

- p_{op} , le processus opératoire sur lequel travaille le SMA de conception.
- T^{a_i} , la valeur de la tolérance au changement (température) de l'agent-flot.
- pa^{a_i} , la valeur du palier de tolérance de l'agent-flot, c'est à dire l'intervalle durant lequel l'agent-flot gardera une tolérance au changement constante.
- $i_{crt}^{a_i}$, l'îlot d'assemblage courant. Un flot d'assemblage i^k est décrit par le quadruplet:

$i^k = \langle t_{ci}^k, p_i^k, C_i^k, T_i^k \rangle$ avec:

- t_{ci}^k , le temps de cycle de l'îlot d'assemblage;
 - p_i^k , le processus d'assemblage sur la base duquel l'îlot a été généré;
 - C_i^k , l'ensemble des cellules de l'îlot d'assemblage;
 - T_i^k , l'ensemble des systèmes de transfert entre les cellules de l'îlot d'assemblage;
- $i_{new}^{a_i}$, l'îlot d'assemblage qui pourra constituer le nouvel état de $i_{crt}^{a_i}$,
 - $i_{opt}^{a_i}$, l'îlot d'assemblage optimal du moment.
 - $cs_{tech}^{a_i}$, valeur booléenne indiquant si la condition de satisfaction technologique est remplie ou non

B.4.4 Comportement

Le comportement de l'agent-flot est de type réactif et il est guidé par la recherche de satisfaction: la réception d'un message en provenance d'une accointance déclenche une réaction appropriée, afin de se rapprocher des conditions de satisfaction.

Condition de satisfaction

Les conditions de satisfaction de l'agent-flot sont:

- *Condition de satisfaction technologique*: l'instanciation de $i_{new}^{a_i}$ constitue un îlot d'assemblage complet, couvrant l'ensemble du processus opératoire p_{op} ($cs_{tech}^{a_i} = True$).
- *Condition de satisfaction de parcours*: l'agent-flot a effectué une trajectoire complète dans le sous-espace physique, déterminé par le processus opératoire ($T^{a_i} = 0$).

L'agent-flot n'a pas de condition de satisfaction d'intégration.

message	expéditeur	contenu	réaction
<i>Modified</i>	a_c^k	-	révision
<i>Dead</i>	a_c^k	-	révision
<i>New Search</i>	a_u	-	activation
<i>Process</i>	a_p	$p_{crt}^{a_p}$ ou $p_{opt}^{a_p}$	mise à jour

Tableau B.7: Conditions de déclenchement des réactions de l'agent-flot avec: a_u , l'utilisateur; a_p , l'agent-processus; $p_{crt}^{a_p}$, le processus courant de a_p ; $p_{opt}^{a_p}$, le processus optimal de a_p ; a_c^k , un agent-cellule

Réactions

La réaction déclenchée par l'agent-flot dépend du message reçu. Le tableau B.7 donne les conditions de déclenchement des réactions de l'agent-flot.

La réaction de révision est liée à la génération d'un flot d'assemblage. Les réactions d'activation et de mise à jour sont liées au parcours de l'espace physique.

Réaction de révision La réaction de révision permet à l'agent-flot de se rapprocher de sa condition de satisfaction technologique. Elle est déclenchée par un message de modification ("Modified") ou un avis de décès ("Dead") en provenance d'un agent-cellule. Si la condition technologique est remplie, l'agent-flot se rapproche de sa condition de parcours en appliquant le recuit simulé et en actvant la génération d'un nouvel flot.

La réaction de révision se déroule de la manière suivante:

1. *Mise à jour des cellules de l'ilot*: l'agent-flot remplace dans C_i^{new} la cellule concernée par la cellule modifiée, ou si c'est une nouvelle cellule, il la rajoute à C_i^{new} . Si l'agent-cellule s'est suicidé, l'agent-flot enlève la cellule concernée de C_i^{new} .
2. *Contrôle de la satisfaction technologique*: après avoir modifié $i_{new}^{a_i}$, l'agent-flot vérifie que l'ensemble des p_c^k , les parties de processus exécutées dans les cellules, couvrent le processus complet de p_{op} . Si c'est le cas, $cs_{tech}^{a_i} = True$.
Remarque: la condition de cohérence, si elle est vérifiée, indique implicitement que tous les autres agents $\in A_{mc}$ sont satisfaits.
3. *Création des équipements de transfert*: si $cs_{tech}^{a_i} = True$, l'agent-flot complète la spécification de l'ilot d'assemblage par la création des équipements de transfert entre les cellules, et son temps de cycle⁴ p_i^{new} est identifié à p_{op} .
4. *Evaluation de l'ilot d'assemblage*: si $T^{a_i} = 0$, la condition de parcours est remplie et l'agent-flot ne fait plus rien, sinon il applique le recuit simulé.

La probabilité d'acceptation de $i_{new}^{a_i}$ est:

$$a(T^{a_i}) = \begin{cases} 1 & \text{si } f_c(i_{new}^{a_i}) \leq f_c(i_{crt}^{a_i}) \\ \exp\left(\frac{f_c(i_{new}^{a_i}) - f_c(i_{crt}^{a_i})}{T^{a_i}}\right) & \text{si } f_c(i_{new}^{a_i}) > f_c(i_{crt}^{a_i}) \end{cases}$$

⁴Pour plus de détails, voir l'annexe D.3.

Si $i_{new}^{a_i}$ est accepté, il devient $i_{crt}^{a_i}$. Si $f_c(i_{crt}^{a_i}) < f_c(i_{opt}^{a_i})$ alors $i_{crt}^{a_i}$ devient $i_{opt}^{a_i}$.

L'agent-flot met alors à jour sa tolérance au changement: si $pa^{a_i} = 0$, $pa^{a_i} = pa_{init}$ et $T^{a_i} = T^{a_i} - a$, sinon $pa^{a_i} = pa^{a_i} - 1^5$. Il se rapproche ainsi de sa condition de satisfaction de parcours.

$i_{new}^{a_i}$ est remis à zéro pour pouvoir générer un nouvel flot.

5. *Réactivation du SMA*: si $T^{a_i} > 0$, l'agent-flot envoie à chaque agent-poste $\in A_p$ un message de réactivation ("Reactivate") contenant la probabilité de rejet pour les agents-postes T^{a_i}/T_{init} . Ceci aura pour conséquence le rejet de certains agents-tâches par les agents-postes et le système va tendre vers une nouvelle solution.

Réaction d'activation La réaction d'activation est déclenchée par la réception d'un message, en provenance de l'utilisateur, requérant un nouveau parcours de l'espace physique ("New Search"). Ce message a pour conséquence la perte de la satisfaction technologique et de la satisfaction de parcours. L'agent-flot, en réaction, demande un nouveau processus d'assemblage, afin de pouvoir effectuer un nouveau parcours et se rapprocher de ses conditions de satisfaction.

La réaction d'activation se déroule de la manière suivante:

1. *Mise à jour des connaissances*: l'agent-flot initialise ses paramètres du recuit simulé ($T^{a_i} = T_{init}$ et $pa^{a_i} = pa_{init}$), et modifie ses connaissances ($cs_{tech}^{a_i} = False$ et $i_{crt}^{a_i} = i_{new}^{a_i} = \{\emptyset\}$).
2. *Requête d'un processus d'assemblage*: l'agent-flot envoie une demande de processus optimal de l'agent-processus ("Process Required") et se met en attente de la réponse.

Réaction de mise à jour La réaction de mise à jour de la population d'agents-tâches est déclenchée à la réception d'un nouveau processus d'assemblage en provenance de l'agent-processus ("Process"). Elle a pour conséquence la création du processus opératoire sur lequel va travailler le module de conception, et la création et la disparition d'agents-tâches.

La réaction de mise à jour se déroule de la manière suivante:

1. *Transformation du processus d'assemblage*: le processus d'assemblage reçu de l'agent-processus et composé de tâches constitutives (géométriques, physiques et complémentaires), est complété par des tâches positionnelles de chargement, de déchargement et de réorientation:
 - une tâche de chargement est ajoutée au processus entre le point de distribution d'un récepteur et le point de travail de la première tâche qu'il subit, si il n'existe pas de tâche avant le point de distribution du récepteur;
 - une tâche de déchargement est ajoutée au processus après la dernière tâche d'assemblage du processus;

⁵Les valeurs des paramètres du profil de refroidissement du module de conception sont les suivantes: $T_{init} = \frac{n}{3} * 41$, avec n le nombre de tâches dans p_{op} et 41, le coût moyen d'un poste mono-tâche; $pa_{init} = 10$; $a = 20$.

- une tâche de réorientation est insérée dans le processus entre deux tâches constitutives successives, si l'orientation du sous-assemblage commun aux deux tâches est différente en sortie de la tâche amont de celle en entrée de la tâche aval.

Le processus opératoire ainsi obtenu constitue p_{op} , le processus sur lequel va travailler le système de conception.

2. *Comparaison avec l'ancien processus opératoire*: l'agent-flot compare le nouveau processus p_{op} avec l'ancien processus sur lequel travaillait le SMA de conception, afin de déterminer deux sous-ensembles de tâches:

- T_{new} , l'ensemble des nouvelles tâches apparues dans p_{op} .
- T_{kill} , l'ensemble des tâches qui ont disparu de p_{op} .

3. *Mise à jour des agents-tâches*: pour chaque tâche de T_{new} (tâche constitutive ou positionnelle), l'agent-flot crée un agent-tâche. Cet ensemble d'agents-tâches, $A_i^{T_{new}}$, est ajouté à la population des agents-tâches: $A_i^{p_{op}} = A_i^{p_{op}} \cup A_i^{T_{new}}$. L'agent-flot détermine également l'ensemble des agents-tâches, $A_i^{T_{kill}}$, qui correspondent aux éléments de T_{kill} .

4. *Activation des agents-tâches*: l'agent-flot envoie à chaque agent-tâche de $A_i^{T_{new}}$ un message d'activation ("Activate") et à chaque agent-tâche de $A_i^{T_{kill}}$ un message de suicide ("Suicide").

Information diffusée

L'information diffusée regroupe les décisions prises par l'agent-flot durant ses réactions. Le tableau B.8 résume les messages envoyés par l'agent-flot.

message	destinataire	contenu	réaction
<i>Reactivate</i>	a_p^k	T^{a_i}/T_{init}	révision
<i>Process Required</i>	a_p	Optimal	activation
<i>Activate</i>	a_i^k	-	mise à jour agent-tâches
<i>Suicide</i>	a_i^k	-	mise à jour agent-tâches

Tableau B.8: Messages envoyés par l'agent-flot selon ses réactions avec: a_i^k , un agent-tâche; a_p^k , un agent-poste.

Annexe C

Calcul de la difficulté d'une tâche d'assemblage

La difficulté estimée d'exécution d'une tâche d'assemblage t^k est donnée par la fonction $d_i^k = \sigma^k + \alpha^k + \pi^k + \rho^k$ avec:

- σ^k , le facteur caractérisant la stabilité du récepteur dans le sous-assemblage concerné. Sa valeur est comprise entre 1 et 10:
 - si la caractéristique physique qui fixe la caractéristique géométrique d'assemblage rattachant le récepteur au sous-assemblage, est réalisée: $\sigma^k = 1$
 - sinon: $\sigma^k = 10 - 0.75n$ avec n , le nombre de demi-degrés de liberté du récepteur bloqués. $n \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$.
- α^k , le facteur caractérisant l'accessibilité au récepteur dans le sous-assemblage concerné. Sa valeur, comprise entre 1 et 10, dépend de la présence d'obstacles dans le volume dont la base est le récepteur, et s'étendant au dessus de lui, selon la direction de travail de la tâche: $\alpha^k = 1 + 1.5n$ avec n , le nombre de demi-degrés de liberté en translation ou en rotation autour de l'axe d'approche pouvant être bloqués par l'obstacle. $n \in \{0, 1, 2, 3, 4, 5, 6\}$.
- π^k , le taux de parallélisation. Ce facteur caractérise le rapport entre la profondeur minimale des deux branches du processus effectuées en parallèle avant la réalisation de la tâche t^k , et la profondeur maximale que ces branches peuvent avoir. Il marque aussi si t^k est une source de parallélisation pour le processus: $\pi^k = 10 - 9\tau$ avec

$$\tau = \begin{cases} \text{si } n_i^k \leq 1 & 0 \\ \text{si } n_i^k \text{ est pair} & \frac{\min(n_i^i, n_i^j)}{n_i^k/2} \\ \text{si } n_i^k \text{ est impair} & \frac{\min(n_i^i, n_i^j)}{(n_i^k-1)/2} \end{cases}$$

n_i^x désigne le nombre de tâches déjà effectuées dans le sous-assemblage Σ^x : k désigne le sous-assemblage résultant de la réalisation de t^k et i et j désignent les sous-assemblages en entrée de t^k .

- ρ^k , la valeur caractérisant l'utilisation des récepteurs entre la tâche t^k et sa tâche aval t^{aval} dans le processus. La valeur de ρ^k est déterminée au moyen des règles présentées dans la figure C.1. Si la valeur est égale à 4 ou 5.5, les circonstances impliquent l'introduction d'opérations supplémentaires d'orientation lors du découpage fonctionnel de t^k .

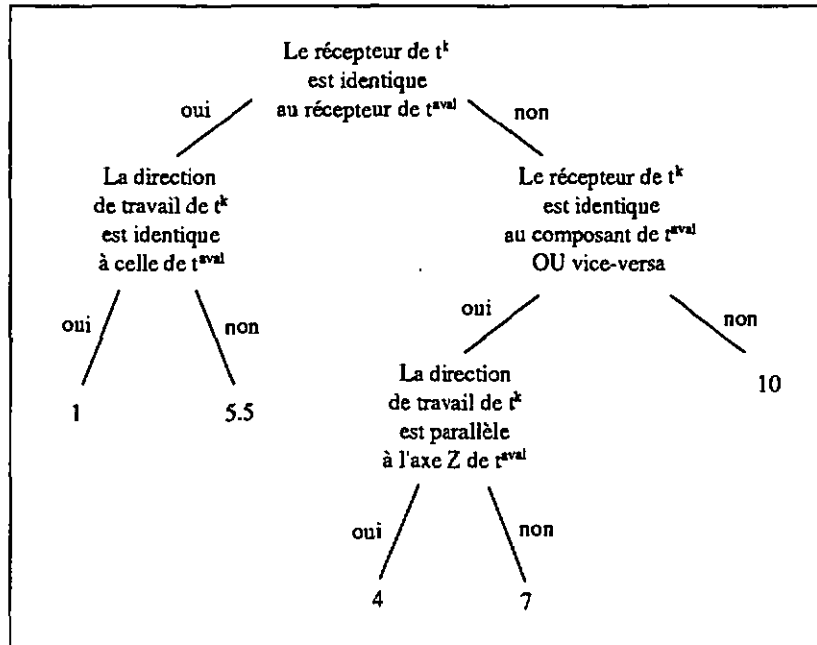


Figure C.1: Détermination de la valeur de ρ^k

Annexe D

Détermination des équipements

D.1 Création des ressources génériques

La création des ressources génériques par un agent-tâche est constituée de trois étapes:

1. *Détermination des points d'entrée et de sortie:* Dans le cas des tâches unaires (physiques, complémentaires, chargements, déchargements, réorientations) la tâche possède un point d'entrée caractérisé par le sous-assemblage récepteur (composition et orientation), et un point de sortie caractérisé par le sous-assemblage résultat (composition et orientation).

Dans le cas des tâches binaires (géométriques) la tâche possède deux points d'entrée - un correspondant au sous-assemblage récepteur (composition et orientation) et un correspondant au sous-assemblage composant (composition et orientation) - et un point de sortie caractérisé par le sous-assemblage résultant (composition et orientation).

2. *Détermination des points N0 et N1:* Les points N0 (lieu de la transformation effective d'un sous-assemblage) sont déterminés sur la base des points d'entrée et de sortie de la tâche: si l'orientation du sous-assemblage minimal (plus petit sous-assemblage commun à deux points) est identique à l'un des points d'entrée et au point de sortie et qu'il existe une tâche amont au point d'entrée, alors ce point d'entrée et le point de sortie sont combinés pour donner un point N0 caractérisé par le sous-assemblage minimal, son orientation et la tâche. Si ce n'est pas le cas, seul le point de sortie constitue le point N0.

Les points N1 (lieu de distribution des sous-assemblages) sont tous les points d'entrée qui n'ont pas été combinés. Ils sont caractérisés par leur sous-assemblage (composition et orientation) et la tâche.

3. *Création des ressources génériques:* Pour chaque point N0 et N1 de la tâche, l'agent-tâche crée une ressource générique de maintien ("posage") caractérisée par le sous-assemblage (composition et orientation), et le point pour lequel elle a été créée.

Pour chaque point N1 de la tâche, l'agent-tâche crée une ressource générique de transfert ("opérateur") entre le point N1 et le point N0 de la tâche. Un opérateur est caractérisé par les deux points qu'il relie et le sous-assemblage minimal des deux points (composition).

L'agent-tâche crée également un opérateur entre les points d'entrée (N0 ou N1) de la tâche à réaliser et le point N0 de (des) la tâche(s) précédente(s).

D.2 Création d'un poste

La création d'un poste d'assemblage par un agent-poste se fait par intégration des ressources génériques et sélection des équipements. Elle se déroule de la manière suivante:

1. *Intégration des équipements spécialisés*: Pour les équipements spécialisés (presse, poste à souder, dispenseur de colle, doigt de maintien...), l'agent-poste s'adapte aux contraintes qu'ils représentent: séquence d'actions de l'équipement spécialisé, nécessité d'un posage lié à l'équipement spécialisé, système fixe ou embarqué, possibilité de coordination avec d'autres opérateurs ou non.
2. *Choix des distributeurs*: Pour chaque ressource générique N1, l'agent-poste détermine un distributeur. Ce choix se fait en fonction du conditionnement du sous-assemblage concerné:

Conditionné en vrac:	bol vibrant
Conditionné en bande:	distributeur en bande
Conditionné en barre:	chargeur
Conditionné en plateau alvéolé:	table simple ou table XY

Selon les circonstances, l'orientation du sous-assemblage en sortie du distributeur est remise à jour.

3. *Choix des posages N0*: Si les circonstances le permettent, l'agent-poste intègre les ressources génériques N0 des agents-tâches dans un même équipement. Les circonstances sont: il n'y a pas de contraintes dues aux équipements spécialisés, l'accessibilité au posage est garantie (nombre d'opérateurs autour du posage), le sous-assemblage minimal (composition et orientation) est identique pour chaque posage N0 générique.
Si les conditions ne sont pas remplies, il génère un nouvel équipement pour la ressource générique N0 concernée.
4. *Détermination des opérateurs*: Les distributeurs et les posages N0 étant déterminés, l'agent-poste détermine les opérateurs et leurs préhenseurs de manière à minimiser le coût: il examine toutes les combinaisons possibles d'opérateurs et retient la solution la moins chère.

La description du poste généré est complétée par son temps de cycle. Le temps de cycle du poste se calcule de la manière suivante:

- Pour chaque équipement N0, le temps de cycle est:

$$t_c^{N0} = \sum_{n=1}^{n_{op}} t_u^n + \max(t_m^1, \dots, t_m^{n_{op}}) \text{ avec}$$

- n_{op} , le nombre d'opérateurs et d'équipements spécialisés qui dépendent de l'équipement NO
 - t_u^n , le temps utile du n-ième opérateur (ou équipement spécialisé)
 - t_m^k , le temps de manipulation du k-ième opérateur
- Le temps de cycle du poste vaut alors:

$$t_{cp}^j = \max(t_c^{N0_1}, \dots, t_c^{N0_i}, t_t^j) \text{ avec}$$
 - $t_c^{N0_k}$, le temps de cycle du k-ième équipement NO
 - i , le nombre d'équipements NO dans le poste
 - t_t^j , le temps de transfert entre les postes de la cellule dans laquelle est intégré le poste.

D.3 Création de l'îlot

La création d'un îlot d'assemblage par l'agent-flot se fait par intégration des cellules des agent-cellules. Une fois que l'ensemble des cellules couvre complètement le processus opératoire, l'agent-flot détermine les transferts entre cellules. Ceci se fait en deux étapes:

1. *Détermination des postes de chargement et déchargement*: Pour chaque cellule, l'agent-flot détermine les équipements nécessaires en entrée et en sortie. Selon la méthode de création des postes, il crée les postes de chargement et de déchargement de la cellule, en s'adaptant aux circonstances (orientation et composition des sous-assemblages concernés).
2. *Choix des systèmes de transfert*: En fonction des résultats de l'étape précédente, l'agent-flot choisit le système de transfert et le type de palette nécessaires, de manière à en minimiser le coût. Cette recherche se fait de manière systématique en gardant la solution la moins chère.

La spécification de l'îlot est complétée par son temps de cycle qui vaut: $t_{ci} = \max(t_{cc}^1, \dots, t_{cc}^n)$, le temps de cycle de la cellule la plus lente.