

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Étude et développement d'un système expert conseil pour la mise en peinture

Declercq, Karin; Parache, Jacqueline

Award date:
1986

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

ETUDE ET DEVELOPPEMENT
D'UN SYSTEME EXPERT CONSEIL
POUR LA MISE EN PEINTURE.

Mémoire présenté en vue de l'obtention du
titre de licencié et maître en informatique
par Karin DECLERCQ
et Jacqueline PARACHE.

Promoteur : Monsieur A. van Lamsweerde.

Année académique 1985 - 1986.

"CHERCHER NE SIGNIFIE RIEN

EN PEINTURE.

CE QUI COMPTE,

C'EST TROUVER..."

P. Picasso.

Remerciements.

Nous remercions en premier lieu Monsieur Axel van Lamsweerde, professeur aux Facultés Notre-Dame de la Paix à Namur et promoteur de ce mémoire, pour les précieuses indications qu'il nous a fournies en cours de rédaction de ce mémoire.

Nous tenons également à remercier la firme TRIMETAL et son personnel pour l'accueil chaleureux que nous y avons reçu, et pour les moyens qui ont été mis à notre disposition pour mener à bien la réalisation de notre projet.

Nous exprimons particulièrement notre gratitude à Monsieur Duvivier, responsable du service informatique, à Monsieur Fraiteur, qui nous a introduites au sein de la firme, et à Messieurs Goossens et Collard, du laboratoire chimique, qui nous ont patiemment fourni les renseignements indispensables dans le domaine du conseil de la mise en peinture.

Enfin, nous remercions toutes les personnes qui, de près ou de loin, ont contribué à l'élaboration de ce mémoire.

TABLE DES MATIERES

TABLE DES MATIERES.

Remerciements.

Table des matières.

Chapitre 1 : Introduction. (K. Declercq, J. Parache)

Chapitre 2 : Cadre général. (K. Declercq)

- 2.1. Enquête sur les systèmes experts de diagnostic.
 - 2.1.1. Classification possible des systèmes experts.
 - 2.1.2. Spécificités d'un système expert de diagnostic.
 - 2.1.3. MYCIN : un exemple de système d'aide au diagnostic.
- 2.2. Travaux déjà effectués sur le sujet.
- 2.3. Contribution d'un système expert au conseil en peinture.

Chapitre 3 : Cadre méthodologique pour la construction d'un système expert. (K. Declercq)

- 3.1. Rappel du cycle de vie traditionnel : software engineering.
- 3.2. Présentation de la démarche de HAYES-ROTH :
 - knowledge engineering.
 - 3.2.1. Identification du problème à traiter.
 - 3.2.2. Conceptualisation.
 - 3.2.3. Formalisation.
 - 3.2.4. Implémentation.
 - 3.2.5. Tests.
- 3.3. Dégagement des spécificités du knowledge engineering : vue personnelle.

Chapitre 4 : Construction de CONSEIL.

- 4.1. Identification du problème à traiter.
 - (K. Declercq, J. Parache)
 - 4.1.1. Identification des participants et de leur rôle.
 - 4.1.1.1. Les commerciaux.
 - 4.1.1.2. Les vendeurs.
 - 4.1.1.3. Les chimistes.
 - 4.1.1.4. Point de vue adopté.
 - 4.1.2. Identification de l'objectif.
 - 4.1.3. Première identification du problème.

TABLE DES MATIERES

- 4.1.4. Nature de l'expertise.
- 4.1.5. Identification des ressources.
- 4.2. Conceptualisation. (J. Parache)
 - 4.2.1. Choix d'un sous-système.
 - 4.2.2. Les concepts.
 - 4.2.2.1. Les supports.
 - 4.2.2.2. Les traitements.
 - 4.2.2.3. La relation "modifie".
 - 4.2.3. Elaboration d'un diagramme des tâches.
 - 4.2.4. Première ébauche des règles contraignant la construction d'un système de peinture.
 - 4.2.4.1. Règles dépendant de la composition chimique.
 - 4.2.4.2. Règles sur les systèmes.
 - 4.2.4.3. Règles portant sur le type de support.
- 4.3. Formalisation. (J. Parache)
 - 4.3.1. Modélisation du problème de mise en peinture.
 - 4.3.2. Processus de raisonnement.
 - 4.3.3. Etude détaillée des composants du modèle.
 - 4.3.3.1. Les supports.
 - 4.3.3.2. Les traitements.
 - 4.3.4. Construction régressive de la suite de traitements.
 - 4.3.4.1. Sélection d'un traitement.
 - 4.3.4.1.1. Principe.
 - 4.3.4.1.2. Synthèse des règles de sélection.
 - 4.3.4.2. Passage au nouveau vecteur d'état.
 - 4.3.4.2.1. Principe.
 - 4.3.4.2.2. Synthèse des règles de modification.

TABLE DES MATIERES

4.4. Implémentation. (J. Parache)

4.4.1. Stratégie globale.

4.4.2. Définition des objets manipulés.

4.4.2.1. Les types d'objets.

4.4.2.1.1. Structure.

4.4.2.1.2. Primitives de manipulation.

4.4.2.2. Les types d'objets génériques.

4.4.2.2.1. Structure.

4.4.2.2.2. Primitives de manipulation.

4.4.3. Structure des représentations internes.

4.4.3.1. Définitions de nouvelles structures PROLOG.

4.4.3.2. Structures de représentation des types d'objets.

4.4.3.3. Structures de représentation des types g

4.4.4. Architecture de la base de faits.

4.4.5. Architecture de la base de règles.

4.4.5.1. Procédures de manipulation de listes.

4.4.5.2. Procédures de manipulation de LISTE_DE_CRITERE.

4.4.5.3. Procédures de manipulation de VECTEUR D'ETAT.

4.4.5.4. Procédures de manipulation de la base de faits.

4.4.5.5. Procédures pour la construction d'une suite de traitements.

4.4.5.5.1. Sélection d'un traitement.

4.4.5.5.2. Calcul du nouveau vecteur d'état.

4.4.5.6. Interface.

4.4.5.6.1. Primitives de programmation de l'interfac

4.4.5.6.2. Interface proprement dite.

Chapitre 5 : Conclusion. (K. Declercq, J. Parache)

5.1. Performances.

5.2. Extensions.

5.3. Voies d'avenir.

Chapitre 6 : Bibliographie.

Chapitre 7 : Annexes.

A. Annexe technique. (K. Declercq)

B. Exemple de fiche technique.

C. Normes.

D. Code du programme.

1. INTRODUCTION

1. INTRODUCTION.

Donner un conseil en peinture à un peintre professionnel ou amateur, c'est lui conseiller la suite de traitements à réaliser sur l'objet qu'il doit peindre de façon à ce que cet objet remplisse certaines exigences après l'application de ces traitements.

Si on appelle système de peinture une séquence d'opérations à effectuer sur un support, et si on considère qu'un tel système est déterminé par un certain nombre d'éléments connus sur l'objet à peindre, la problématique apparaît relativement simple, surtout si on admet la solution "pas de solution" dans l'ensemble des solutions.

Avec le problème présenté de cette manière, il semble normal de proposer une solution basée sur une table de décision à deux entrées, l'une pour l'objet à peindre et l'autre pour le système de peinture. Mais là où le bât blesse, c'est d'une part qu'il existe une centaine de produits, ce qui donne une dizaine de milliers de systèmes de peintures différents, et d'autre part que les combinaisons de critères servant à décrire les objets à peindre se chiffrent par centaines de milliers. De plus, les caractéristiques des systèmes changent dans le temps : un produit peut disparaître, un autre peut apparaître. Dès lors, cet outil se révèle beaucoup trop lourd à l'usage, et l'ensemble de connaissances qu'il peut contenir n'est que partiellement extensible.

C'est la raison pour laquelle le personnel chargé du conseil en peinture a toujours estimé que cette tâche ne pourrait être informatisée dans l'état actuel des techniques informatiques. En particulier, un prototype fonctionnant sur base d'une table de vérité, a permis de faire sentir les limites d'une approche traditionnelle.

Les possibilités de l'Intelligence Artificielle (entre autres les techniques de représentation des connaissances et de recherche dans un espace de solutions et l'apparition dans le monde de l'industrie, de langages de programmation adaptés) ont donné au promoteur de notre projet, l'idée de créer un système conseil qui pourrait seconder le personnel chargé de donner le conseil-peinture ou le remplacer lorsqu'il n'existe pas. Ce système devrait être dépourvu de toute subjectivité par rapport aux produits à conseiller et se bornerait à donner toutes les solutions au problème posé par l'utilisateur, c'est-à-dire tous les systèmes de peinture techniquement corrects; le système amélioré pourrait établir entre ces solutions un ordre de préférence basé sur des critères relevant par exemple de décisions de marketing ou de gestion de stocks.

1. INTRODUCTION

Notre rôle consistait dès lors à prendre connaissance du problème, et à étudier une solution plus dynamique qui devrait aboutir à la réalisation d'un véritable système expert de mise en peinture. Cette recherche avait aussi pour objectif parallèle d'apporter aux constructeurs du premier prototype un point de vue nouveau, tout à fait indépendant de contraintes machine. Au lieu de considérer un système de peinture comme une solution atomique, il s'est avéré plus intéressant de l'envisager comme un ensemble de produits liés par le fait qu'ils apportent un élément de réponse à la question de l'utilisateur.

Nous nous sommes donc penchées sur le problème de la construction d'un prototype du système conseil. Ce dernier doit être évalué sous l'angle de la stratégie de construction du système de peinture et de la validité des solutions données par rapport aux informations dont il dispose. Nous avons accordé moins d'importance aux contraintes de cohérence et de complétude de la base de connaissances.

Dans ce travail, nous avons étudié les problèmes posés par la conception d'un système expert, et abordé la conception pratique d'un prototype dont le domaine est le conseil en peinture. La découpe de notre travail en trois parties principales est assez classique, mais la découpe interne de la deuxième partie l'est peut-être moins.

La première partie situe notre système dans le cadre des recherches en Intelligence Artificielle, et plus précisément dans celui des systèmes experts d'une part, et dans le cadre des travaux entrepris dans le domaine du conseil en peinture. Elle constitue une présentation générale concernant les systèmes experts en diagnostic, et un exposé relatif au premier prototype, construit sur base de tables de décision, au sein de la firme TRIMETAL. Elle expose les types de problèmes qui se posent dans le domaine du conseil en peinture, et souligne la contribution des systèmes experts à la résolution de ces problèmes.

La deuxième partie constitue le coeur de ce travail, et est également constituée de deux chapitres. Le premier d'entre eux se veut une description des principes méthodologiques sous-jacents à la conception de notre système; ces principes sont issus de [HAY83] et [FEI84]. Ces principes feront l'objet d'une description brève, et après avoir rappelé les étapes de la méthode classique de construction d'un logiciel [VAN84], nous esquisserons une comparaison des deux méthodes, basée sur l'expérience que nous en avons. Le deuxième chapitre relate les différentes étapes que nous avons suivies pour développer notre système, parallèlement aux principes théoriques décrits dans le chapitre précédent. Nous y donnerons également une description des difficultés auxquelles nous avons été confrontées, et la façon dont nous les avons résolues.

1. INTRODUCTION

La troisième et dernière partie propose une évaluation du travail réalisé, et les bases pour des travaux ultérieurs. Les travaux ultérieurs qui pourraient constituer une suite logique de ce travail sont suggérés dans ce chapitre, et quelques conseils sont donnés aux personnes qui seraient éventuellement intéressées par leur réalisation. Ces conseils découlent directement de notre expérience.

On trouvera en fin de travail les annexes. La première est une annexe technique qui comprend les bases théoriques de la chimie des peintures et de la philosophie des systèmes de peinture. La deuxième présente un échantillon du matériel dont nous disposons en début d'analyse, la troisième est constituée des normes que nous avons utilisées dans la conception du prototype. La quatrième et dernière contient le code de la base de connaissance et du programme PROLOG.

2.1. ENQUETE SUR LES SYSTEMES EXPERTS DE DIAGNOSTIC.

2. CADRE GENERAL.

Nous allons dans ce chapitre situer notre projet dans le cadre d'une part de la recherche en systèmes experts et d'autre part du monde de la peinture en général et de la firme TRIMETAL en particulier.

2.1. ENQUETE SUR LES SYSTEMES EXPERTS DE DIAGNOSTIC.

Si on considère la définition générique d'un système expert, les systèmes de diagnostic n'ont ni plus, ni moins de caractéristiques que les autres. Rappelons ici ces caractères généraux :

- les systèmes experts doivent être capables de manipuler une grande quantité de connaissances concernant un domaine très circonscrit;
- ils doivent pouvoir communiquer avec leurs utilisateurs dans un sous-ensemble du langage naturel, et donc être pourvus d'une interface assez perfectionnée d'analyse et d'interprétation de phrases;
- ils doivent être capables d'expliquer aux utilisateurs le cheminement du raisonnement qui leur a permis d'arriver à la solution qu'ils donnent;
- enfin, leur expertise doit pouvoir être étendue, et ce idéalement par communication avec des experts non-informaticiens.

Nous avons trouvé dans la littérature une classification des systèmes experts qui nous a paru intéressante.

Après avoir précisé cette classification, nous évoquerons plus particulièrement une de ses classes, dans lesquelles il nous semble que notre système puisse entrer. Cette classe est celle des systèmes experts de diagnostic.

2.1. ENQUETE SUR LES SYSTEMES EXPERTS DE DIAGNOSTIC.

2.1.1. CLASSIFICATION POSSIBLE DES SYSTEMES EXPERTS.

Les chercheurs du centre Xerox de Palo Alto [STE82] ont proposé une classification des tâches génériques exécutées par les experts, basée sur les caractéristiques de ces tâches.

Elle reprend six classes de tâches :

1. l'interprétation, c'est-à-dire l'analyse des données afin d'en découvrir la signification;

2. le diagnostic, c'est-à-dire la découverte d'erreurs (ou de maladies pour un organisme vivant), basée sur l'interprétation de données parfois imprécises;

3. le monitoring, c'est-à-dire l'interprétation continue de signaux et le déclenchement d'une alarme si une intervention extérieure est requise;

4. la prédiction, c'est-à-dire la prévision de l'évolution d'un modèle présent ou passé;

5. le planning, c'est-à-dire la mise au point d'un programme d'actions à exécuter pour atteindre un objectif final;

6. le design, c'est-à-dire la mise au point de spécifications pour créer des objets qui satisfont des exigences particulières.

Ces différentes classes ont pour point commun l'intervention d'un nombre important de paramètres interdépendants, qui sont souvent de nature hétérogène, dont l'observation peut être erronée et dont l'interprétation n'est pas bien codifiée dans les manuels.

On peut donner des exemples de cas entrant dans ces différentes classes :

1. interprétation de données provenant d'un spectographe de masse (DENDRAL);

2. diagnostic de maladies infectieuses du sang (MYCIN);

3. monitoring d'un patient placé sous surveillance respiratoire après une intervention chirurgicale;

4. prédiction des effets d'une réorientation de la politique économique d'une région;

5. expérimentation de plans en génétique moléculaire;
planning des déplacements d'un robot (STRIPS);

6. design de configurations de VAX (R1).

2.1. ENQUETE SUR LES SYSTEMES EXPERTS DE DIAGNOSTIC.

2.1.2. SPECIFICITES D'UN SYSTEME EXPERT DE DIAGNOSTIC.

Le premier type de diagnostic qui vient à l'esprit est le diagnostic médical. Ce n'est pas la seule acception de ce terme : détecter la nature d'une panne de machine, c'est aussi poser un diagnostic. Dès lors, le terme "système expert de diagnostic" ne signifie pas seulement système de diagnostic médical, mais représente un ensemble de systèmes répondant à certaines caractéristiques que nous allons nous efforcer de décrire ici. Un petit tableau exemplatif permettra une illustration des différents concepts mis en oeuvre, pour le cas du diagnostic médical et de pannes de machine.

Le premier point important d'un système de diagnostic est l'état initial du sujet. Le sujet est l'élément de base sur lequel travaille le système. Ce sujet est l'élément du domaine qui fait l'objet du problème. Son état initial est décrit par l'utilisateur. Cette description peut parfois ne pas être exempte d'erreurs et de lacunes. Elle se compose d'un certain nombre de symptômes, c'est-à-dire de caractéristiques, conformes à des critères définis par l'auteur du système.

Le deuxième est l'état final du sujet. Il est assimilable à un état optimal, et est décrit également par un ensemble de caractéristiques répondant à des critères. L'état final est l'état dans lequel le sujet devra se trouver après avoir subi les remèdes conseillés par le système. Dans le cas où le système expert a comme unique fonction d'émettre un diagnostic sans conseiller de thérapie, l'état final se définit comme le "bon" état, ou l'état "de marche", du sujet.

Le troisième point est le diagnostic posé par le système. Ce diagnostic est inféré à partir de l'état initial du sujet et éventuellement de son état final. Pour ce faire, le système dispose de connaissances emmagasinées dans une base de connaissances. Celles-ci relèvent du domaine spécifique du système, et sont mises au point par un ou plusieurs experts.

Le quatrième et dernier point est la thérapie conseillée par le système. Cette thérapie se compose d'un ensemble de remèdes à donner au sujet pour l'amener de son état initial à son état final. Cet ensemble peut prendre ou non la forme d'une séquence. Pour pouvoir conseiller la thérapie, le système se sert également de sa base de connaissances.

Le petit tableau de la page suivante illustre ces quatre points sur deux exemples.

2.1. ENQUETE SUR LES SYSTEMES EXPERTS DE DIAGNOSTIC.

DOMAINE	MEDECINE	PANNES MACHINE
SUJET	patient	machine
ETAT INITIAL = SYMPTOMES	symptômes décrits par le patient ou observés par le médecin	symptômes observés par le réparateur
ETAT FINAL	bonne santé	machine en état de fonctionnement
DIAGNOSTIC	maladie du patient	panne de la machine
REMEDE	soins divers à donner au patient	réparations à entreprendre sur la machine

Décrit à l'aide des quatre points précédents, le problème de diagnostic en général semble facilement soluble par un simple programme de recherche. Cependant, cette approche simpliste a deux failles. La première est que la connaissance n'est pas toujours exacte, ni complète. En effet, l'état initial du sujet peut être décrit de façon partielle et non complètement, et les symptômes observés peuvent être erronés. La seconde est que les remèdes prescrits lors de l'établissement de la thérapie peuvent avoir des effets secondaires non-désirés et qui ne sont pas toujours précisément décrits. On pensera ici, par exemple, aux réactions allergiques provoquées par la combinaison de deux antibiotiques, et non prévues par le médecin. Il faut donc se ménager des possibilités de traiter des connaissances inexactes et de développer des raisonnements plausibles.

Nous allons maintenant voir comment le plus célèbres des SE de diagnostic médical, MYCIN, applique cette théorie.

2.1. ENQUETE SUR LES SYSTEMES EXPERTS DE DIAGNOSTIC.

2.1.3. MYCIN : UN EXEMPLE DE SYSTEME D'AIDE AU DIAGNOSTIC.

MYCIN est un système développé à partir de 1977 par SHORTLIFFE et ses collègues [BUC84]. Il est destiné à établir un diagnostic et une thérapie dans le domaine des maladies infectieuses du sang, à partir de renseignements provenant d'analyses de sang, d'urine ou autre.

Son travail consiste à récolter des renseignements, puis à donner un diagnostic sur la bactérie présente chez le patient, et à conseiller enfin une thérapie appropriée.

L'état initial du sujet, qui est ici le patient, est composé de données cliniques sur le patient, telles que son âge, son poids, etc, de données sur l'infection dont il souffre, sur la culture biologique qui a été faite lors de l'examen clinique, et sur l'organisme qui prolifère dans cette culture. Dans la plupart des cas, l'utilisateur ne connaît pas l'identité de cet organisme.

L'état final du sujet est évidemment sa bonne santé, c'est-à-dire la disparition de l'organisme fauteur de trouble.

Le diagnostic émis par MYCIN est l'identité de cet organisme. Pour pouvoir l'inférer, le système a besoin de renseignements en suffisance. Dès lors, il cherchera à compléter ses données en demandant interactivement des précisions à l'utilisateur. Ce diagnostic est posé avec un certain degré de plausibilité.

La thérapie conseillée consiste en un certain nombre de médicaments, donnés avec leur posologie et leurs contre-indications. En cas de mauvaise tolérance du patient à une thérapie donnée, MYCIN est capable de rechercher et de proposer des thérapies parallèles pour la guérison du patient.

D'un point de vue technique, MYCIN est un système basé sur les règles de production, et est écrit en LISP. Chacune de ses règles se présente sous la forme :

SI il est évident que A et B sont vrais,
ALORS conclure qu'il est évident que C est vrai;

une deuxième façon de l'exprimer est :

"condition(s) entraîne(nt) action".

A cette forme sont associés des coefficients de plausibilité. En effet, les actions de MYCIN sont des conclusions sur l'état du monde réel, qui n'est pas connu avec certitude. Dès lors, la règle se schématise plutôt comme suit :

SI condition A et condition B sont vraies,
ALORS il est évident (coefficient de plausibilité) que
l'action C est vraie.

2.1. ENQUETE SUR LES SYSTEMES EXPERTS DE DIAGNOSTIC.

Donnons un exemple de règle de ce type :

IF: 1) The identity of the organism is not known with certainty, and
2) This current organism and prior organisms of the patient agree
with respect to the following properties : GRAM MORPH

THEN: There is weakly suggestive evidence (.1) that each of them is a
prior organism with the same identity as this current
organism.

MYCIN utilise des coefficients de plausibilité dans le choix des règles. Ces coefficients s'échelonnent entre -1.0 et 1.0. La valeur 1.0 accordée à une proposition signifie la certitude que cette proposition est vraie; la valeur -1.0, la certitude qu'elle est fausse, et la valeur 0.0 l'indifférence, c'est-à-dire que le système ne voit aucun lien entre les conditions et l'action.

MYCIN utilise dans son moteur d'inférence le chaînage arrière, qui consiste à prouver un objectif en prouvant ses sous-objectifs, jusqu'à obtention de la preuve d'un fait, qui est immédiate. Dès lors, dans un système qui manipule des connaissances inexactes (c'est-à-dire des connaissances associées à un coefficient de plausibilité), cette méthode de recherche conduit à une exploration en profondeur d'abord de l'arbre ET/OU des objectifs. Cette exploration est exhaustive : en effet, il faut trouver toutes les possibilités de prouver un sous-objectif avec leurs coefficients de plausibilité respectifs, pour déterminer celle qui est "la plus plausible", "la plus sûre". Cette recherche est très coûteuse en temps, et constitue un luxe qu'on ne peut se permettre : la base de connaissances croît avec le temps, et une recherche exhaustive devient infaisable.

Pour résoudre ce problème, la recherche est guidée par des méta-règles, c'est-à-dire des règles traitant des règles, qui fournissent une stratégie pour réordonner les règles avant leur appel. Il s'agit donc de créer des règles permettant de modifier l'ordre des sous-butis examinés de façon à gagner du temps. Voici un exemple de sous-règle :

IF : 1) the infection is a pelvic abscess, and
2) there are rules which mention in their premise
ENTEROBACTERACEAE, and
3) there are rules which mention in their premise gram-positive
rods,

THEN : there is suggestive evidence(.4) that the former should be
done before the latter.

Cette méta-règle signifie que si on traite un abcès pelvique, les organismes de la classe des entérobacteriaceae devraient être considérés avant les germes gramme-positif.

Les méta-règles ont deux utilités générales, qui sont soit de créer un ordre partiel dans les sous-objectifs à prouver, soit de renseigner le système sur l'utilité de prouver certains objectifs. Par contre, elles ne disent rien sur la validité de ces règles.

2.1. ENQUETE SUR LES SYSTEMES EXPERTS DE DIAGNOSTIC.

Ces méta-règles sont considérées par le système au moment de prouver un objectif. Le système les consulte d'abord pour déterminer si une d'elles est relative à cet objectif. Si c'est le cas, elle est considérée d'abord, et le système peut obtenir un certain nombre de renseignements sur l'utilité ou l'ordre relatif des sous-objectifs à prouver. Vues en termes d'arbres, les méta-règles ont pour effet de réduire une partie de l'arbre ou de réordonner les branches de cet arbre.

La partie du système qui recherche la thérapie a été écrite par CLANCEY [CLA84], et sa conception a débuté en 1978. Le système se base sur la description des infections du patient, les organismes qui les causent, et un classement préférentiel des médicaments en usage. Le système tient compte des éventuelles allergies du patient, et des combinaisons, séquences et doses inappropriées des médicaments conseillés. En d'autres mots, il vérifie lui-même la cohérence des solutions qu'il propose.

Les mécanismes d'acquisition et de modification des connaissances, ainsi que celui d'explication, sont facilités par un système de traduction de règles de la forme LISP vers l'anglais et vice-versa.

2.2. TRAVAUX DEJA EFFECTUES SUR LE SUJET

2.2. TRAVAUX DEJA EFFECTUES SUR LE SUJET.

La firme TRIMETAL a, par le passé, développé un prototype écrit en FORTRAN pour l'aide au conseil en peinture. Ce prototype travaille sur base de connaissances peu raffinées. Il conseille des systèmes de peinture entiers, et typiques. Dès lors, il ne fournit généralement qu'une seule solution par problème posé, et cette solution, tout en étant techniquement exacte, n'en est pas moins dictée par des critères commerciaux; elle n'est donc pas nécessairement la "meilleure". De plus, les cas exceptionnels ne trouvent pas de solution.

Nous nous proposons de décrire le principe du fonctionnement de ce prototype.

La firme a adopté une classification figée des objets à peindre et une représentation figée et de bas niveau de leurs caractéristiques, qui est la suivante :

- objet codé en 4 caractères,
- matière codée en 2 caractères,
- protection codée en 2 caractères,
- état codé en 2 caractères,
- aspect final codé en 2 caractères.

Dès lors, le nombre d'objets susceptibles d'être décrits de cette manière est strictement combinatoire. A chaque combinaison correspond un code, et à chaque code est associé le code de la solution finale. Les solutions appartiennent à deux classes : les finitions ou les préparations.

Une solution finale est codifiée en 12 caractères, au moyen de la sélection des actions qui la composent. Le fichier des solutions a donc un aspect tel que celui-ci :

```
code sélection  
code sélection  
action 1
```

```
code sélection  
code sélection  
code sélection  
action2
```

Selon le principe d'une table de décisions, on explore un par un les codes sélections. Si l'un d'entre eux correspond au code construit à partir des caractéristiques de l'objet données par l'utilisateur, l'action qui le suit directement est exécutée; sinon, l'exploration se poursuivra aux codes sélections suivants.

Les concepteurs de ce programme ont prévu une facilité, qui consiste à donner un signe particulier dans un code sélection, signifiant que n'importe quel caractère admis dans les codes peut se trou-

2.2. CONTEXTE HISTORIQUE.

ver à cette position. Cette astuce permet de réduire le nombre de codes sélections avant une même action, mais constitue un point faible de l'implémentation.

Un principe semblable est utilisé dans les menus afin de filtrer les réponses devenues impossibles du fait d'une sélection particulière dans un menu précédent.

Nous présentons l'un des systèmes de peinture de TRIMETAL en annexe afin de donner au lecteur une petite idée de la complexité de la codification de ce genre de données.

Des points de vue représentation et manipulation des connaissances, la codification du programme en fait un outil particulièrement lourd à faire évoluer, et rigide.

2.3. CONTRIBUTION D'UN SYSTEME EXPERT AU CONSEIL EN PEINTURE.

2.3. CONTRIBUTION D'UN SYSTEME EXPERT AU CONSEIL EN PEINTURE.

La contribution d'un système expert dans le problème du conseil en peinture se marque sur plusieurs plans.

L'un des reproches que l'on faisait au prototype FORTRAN était le nombre combinatoire de situations envisagées. En effet, le prototype FORTAN tient compte de toutes les combinaisons possibles des cinq caractéristiques du support, par le biais de la construction d'un code sélection propre à chaque combinaison. Dès lors, une augmentation du nombre de valeurs d'une des caractéristiques signifie un changement complet de tout le codage : il faut reclasser les nouveaux codes sélections dans le fichier des solutions et attribuer à chacun l'action qui lui convient; en outre, il faudra peut-être prévoir un caractère supplémentaire pour le code de cette caractéristique, ce qui décale toute la construction du code sélection. C'est la représentation des connaissances qui entraîne ce problème. Au niveau de la manipulation, le prototype fonctionne séquentiellement : après obtention du code sélection, il parcourt le fichier des solutions pour trouver le système qui lui correspond. Il s'agit d'un processus classique dans le traitement des tables de décision. Notre système considère les états du support par rapport à leurs composantes, et ne prévoit pas toutes les situations possibles. Il analyse l'état donné, et oriente ses recherches de solutions en fonction des résultats de cette analyse, et non en fonction d'une table de vérité qui, à chaque situation possible, associe un système de peinture prévu à cet effet. Dans le cas décrit plus haut, aucun changement n'aura lieu dans la partie manipulation de la connaissance. Dans la partie base de connaissance, on veillera à ajouter la nouvelle valeur dans la liste des valeurs possibles de la caractéristique, et à décrire cette valeur à l'emplacement réservé à cet effet.

Le deuxième reproche à faire à ce prototype découle du premier. S'il est difficile de prévoir toutes les situations possibles au niveau du supports, il est aussi ardu de construire tous les systèmes de peinture possibles. Le prototype fonctionne avec une certaine variété de systèmes, lui permettant de couvrir toutes les situations prévues au niveau du support, mais il ne donne qu'une seule solution à un cas précis, alors que la plupart du temps, plusieurs solutions sont envisageables. L'impossibilité de donner plusieurs solutions pourrait être contournée par l'ajout des systèmes de peinture alternatifs, et des codes sélections qui leur correspondent. Du point de vue manipulation des connaissances, le système serait obligé de parcourir tout le fichier des solutions, puisqu'un même code sélection pourrait se trouver à plusieurs endroits. Malgré ces modifications, le système resterait incomplet, vu le nombre énorme de systèmes de peinture à envisager. Notre système bâtit la suite résultat pas à pas, c'est-à-dire traitement par traitement. Comme le langage dans lequel il est écrit est pourvu d'un mécanisme de backtracking, il peut revenir sur le choix d'un traitement pour en considérer un autre, et continuer sa recherche au départ de cette nouvelle ébauche de séquence. C'est sur ce mécanisme que nous nous basons pour pouvoir donner toutes les solutions envisageables au problème posé.

2.3. CONTRIBUTION D'UN SYSTEME EXPERT AU CONSEIL EN PEINTURE.

La connaissance qui se trouve dans le prototype FORTRAN est fortement dépendante du code dans lequel elle se dissimule. Le programme ne comporte pas de base de connaissances spécifiques séparée du bloc de traitement de la connaissance. CONSEIL possède une base de connaissance séparée des modes de manipulation qui permettent de la traiter. De ce fait, l'expertise du programme peut être facilement modifiée sans qu'il soit nécessaire de modifier le code du mécanisme de raisonnement. Ceci était aussi un reproche cité pour le prototype FORTRAN. Dès lors, les ajouts et les retraites de cas possibles ne causent pas de modifications du code, et peuvent faire l'objet d'une interface de modification des connaissances utilisable par un non-informaticien, voire par l'expert du domaine.

Nous allons illustrer ce dernier point par un exemple basé sur la fiche technique qui se trouve en annexe A.

Imaginons que la firme décide de supprimer le produit PLOMOFERRINE, qui fait double emploi avec le produit TRIMINIUM. Ce dernier n'est compatible avec la PERMALINE que moyennant une couche intermédiaire de PRIMERSURFACER. Voyons les modifications à faire dans la représentation du prototype et dans celle de notre système.

Dans le prototype, il faut d'abord modifier tous les systèmes qui mentionnent le produit PLOMOFERRINE, et remplacer ce produit par TRIMINIUM. Ensuite, il faut inclure entre la couche de TRIMINIUM et celle de PERMALINE s'il y en a une, une couche de PRIMERSURFACER.

Il faut aussi s'assurer que l'on n'a pas de systèmes en double, et, le cas échéant, regrouper les codes sélections correspondant à un même système.

Enfin, il faut modifier dans le traitement de la table toutes les instructions qui concernent la PLOMOFERRINE, et les adapter pour le TRIMINIUM.

Dans notre système, il suffit de supprimer dans la base de connaissance toutes les règles qui concernent ce produit, à savoir toutes celles qui contiennent le mot "PLOMOFERRINE", et ajouter les caractéristiques qui concernent le TRIMINIUM. Cette dernière opération sera particulièrement aisée si le système est muni d'une interface de modification de la base de connaissance. Il n'y a absolument rien à modifier dans le mécanisme de raisonnement.

3.1. RAPPEL DU CYCLE DE VIE TRADITIONNEL.

CHAPITRE 3 : CADRE METHODOLOGIQUE POUR LA CONSTRUCTION D'UN SYSTEME EXPERT.

Nous nous proposons de rappeler les étapes du cycle de vie traditionnel d'un logiciel (= software engineering) [VAN84], et d'établir le parallèle avec la méthode de HAYES - ROTH (= knowledge engineering), développée dans [HAY82]. Cette comparaison est une discussion purement personnelle, sans aucune autre prétention.

3.1. RAPPEL DU CYCLE DE VIE TRADITIONNEL : SOFTWARE ENGINEERING.

La démarche énoncée dans [VAN84] comporte cinq étapes.

La première consiste à analyser les besoins de l'organisation à laquelle est destinée le logiciel. Le résultat de cette étape est une première description du problème à informatiser.

La deuxième étape est l'analyse fonctionnelle du système, c'est-à-dire les spécifications fonctionnelles du système automatisé. Cette étape a pour résultat une première définition des données à traiter, des résultats à produire, des traitements à réaliser pour passer des données aux résultats et des interfaces homme / machine.

La troisième étape est le design de conception, et se déroule en deux phases : la conception de l'architecture logicielle du système, à savoir l'identification et la définition des composants de base et de leurs interrelations, et la conception de l'architecture physique du système, à savoir la détermination des algorithmes théoriques et le choix d'une représentation adéquate pour les structures de données.

La quatrième étape consiste à coder les algorithmes théoriques développés à l'étape précédente au moyen des outils informatiques choisis dans la deuxième partie de cette étape.

La dernière étape est le test des composants du système d'abord, et de leur intégration ensuite, et se termine par le test du système complet.

3.2. PRESENTATION DE LA DEMARCHE DE HAYES - ROTH :
KNOWLEDGE ENGINEERING.

La démarche que nous avons suivie est celle proposée dans [HAY82] pour la construction d'un système expert.

Elle comporte cinq phases:

- l'analyse du problème à traiter,
- la conceptualisation de la réalité,
- la modélisation de la réalité,
- l'implémentation,
- les tests.

L'analyse du problème consiste à identifier les caractéristiques du monde réel dans lequel on se propose de travailler. Il s'agit des connaissances, des experts impliqués dans le système et des ressources utilisées, ainsi que de l'objectif et de la spécification du problème à traiter.

La conceptualisation consiste à trouver les concepts qui représentent les objets de la réalité décrits dans la spécification établie lors de la première phase.

La modélisation consiste à trouver un ou des modèles susceptibles de représenter la connaissance définie et conceptualisée dans les deux phases précédentes.

L'implémentation consiste à coder dans le langage choisi les connaissances traitées aux trois phases précédentes, et à écrire les règles qui permettront de les manipuler pour en déduire ou en inférer d'autres.

La phase de test consiste à valider les règles qui organisent la connaissance.

Ce chapitre a pour but de présenter brièvement ces principes méthodologiques que nous avons suivis pour la construction de notre système CONSEIL.

3.2.1. IDENTIFICATION DU PROBLEME A TRAITER.

L'identification du problème à traiter se déroule en quatre phases. La première consiste à décrire l'objectif du système, la deuxième à déterminer quels seront les experts interrogés, et quel sera leur rôle, la troisième à identifier le problème posé, et la dernière à décrire et quantifier les ressources utilisées.

L'identification de l'objectif du système et celle du problème à traiter se font parallèlement. Il est cependant utile de bien saisir la différence des points de vue. L'objectif du système ne concerne pas le domaine. Il s'agit de décrire l'utilité du système dans l'endroit où il sera implanté. Des exemples d'objectifs peuvent être la formalisation d'un ensemble de pratiques, la distribution d'une expertise rare, l'amélioration des méthodes de résolution de l'expert, ou encore l'automatisation des aspects routiniers de son travail. L'identification du problème concerne étroitement le domaine, et est le fruit d'échanges de vues informels entre l'expert du domaine et l'informaticien. Ils aborderont diverses questions, citées ci-dessous, dont le but est de cerner le problème et les structures de connaissances sous-jacentes pour que le développement de la base de connaissances puisse commencer.

Les questions suivantes seront abordées dans cette phase :

- quelle classe de problèmes le système expert va-t-il résoudre ?
- comment ces problèmes peuvent-ils être caractérisés ou définis ?
- quels sont les sous-problèmes importants, et quelle est la répartition des tâches qui en découle ?
- quelles sont les données ?
- quels sont les termes importants et leurs interrelations ?
- à quoi ressemble une solution, et quels sont les concepts qui interviennent ?
- quels sont les aspects essentiels de l'expertise humaine dans la résolution de ces problèmes ?
- quelles sont la nature et l'étendue de la "connaissance pertinente" qui sous-tend les solutions humaines ?
- quelles sont les situations qui semblent empêcher une solution ?
- comment ces obstacles vont-ils affecter le système expert ?

L'interrogation des experts n'est pas toujours aisée. Il faut s'y prendre de façon à ne pas les brusquer, et à les intéresser au problème. En effet, ils peuvent avoir l'impression que l'informaticien crée son système pour les remplacer. On peut appliquer l'expérience décrite dans [FEI84]. Les entrevues se déroulent comme suit. En premier lieu, l'ingénieur de la connaissance persuadera l'expert de se laisser interroger sur son domaine. Une fois l'expert persuadé, l'ingénieur se familiarisera avec le domaine et son jargon en lisant des ouvrages sur le sujet ou en suivant des cours. Il fixera ensuite son premier rendez-vous avec l'expert. Il lui soumettra un problème ni trop ardu, ni trop simple, et essaiera de saisir son mécanisme de raisonnement. Dès la fin de cette première entrevue, la construction d'une maquette commencera. Elle sera proposée à l'expert lors d'un deuxième rendez-vous, destiné à la corriger et à l'améliorer. La construction du système consistera en une longue suite d'entrevues et de correction de la maquette, jusqu'à obtention du système complet.

PRESENTATION DE LA DEMARCHE DE HAYES - ROTH

La maquette joue un rôle très important. Il est d'ordre psychologique. C'est elle en effet qui déterminera l'"accrochage" de l'expert au projet. Cependant, elle pourrait distraire l'expert de sa tâche, et le faire trop penser aux problèmes propres à l'implémentation.

La phase d'identification des experts et de leur rôle a lieu avant le premier rendez-vous. Elle consiste à étudier à partir des destinataires potentiels du système, quels seront les experts à interroger, et quel sera leur rôle.

Le choix des experts constitue une phase importante. En effet, c'est d'eux que provient la plus grande partie de la connaissance, et il importe de les choisir de façon à avoir l'expertise la plus complète, la plus fine et la plus précise possible. Dès lors, on peut être amené à faire un choix pluri-disciplinaire.

Le rôle des experts est de transférer vers l'informaticien la connaissance nécessaire à la construction du système. Il est intéressant de recouper les renseignements provenant de plusieurs experts de la même discipline, et de confronter chacun avec le résultat de ce recoupement, pour obtenir une connaissance plus fine et plus précise.

Le rôle des informaticiens est de "pousser" les experts à donner leur connaissance et à formuler explicitement les mécanismes qu'ils utilisent parfois inconsciemment pour résoudre leurs problèmes. Il faut aussi essayer de leur extraire la connaissance qu'ils ont accumulée durant leur expérience.

Enfin, il importe d'identifier et de quantifier les ressources dont on dispose : sources de connaissances (experts, livres, conférences, expérience professionnelle), temps, argent et matériel informatique.

La connaissance provient de deux classes de sources : celles concernant plutôt l'expert du domaine considéré, et l'autre intéressant davantage l'ingénieur de la connaissance. L'expert se base sur ses propres expériences de résolution, sur la littérature consacrée au domaine, et sur des exemples de résolution de problèmes, pouvant par exemple provenir d'autres experts. L'ingénieur possède une connaissance axée sur son expérience dans le domaine de la construction de systèmes experts et de la résolution de problèmes analogues, et sur les méthodes, les outils et les représentations propres au domaine de l'informatique.

Le temps est une ressource critique. L'expert du domaine et l'ingénieur de la connaissance doivent en consacrer une quantité énorme pour mener leur tâche à bien, et chaque difficulté rencontrée signifie un certain laps de temps supplémentaire pour la surmonter. L'échange de vues entre l'expert du domaine et l'ingénieur de la connaissance soulève énormément de difficultés. En effet, chacun de ces deux experts doit non seulement pénétrer le domaine de l'autre, mais aussi le comprendre suffisamment pour permettre les suggestions mutuelles.

PRESENTATION DE LA DEMARCHE DE HAYES - ROTH

Enfin, l'argent et le matériel informatique sont également des ressources critiques. Les erreurs coûtent cher, les experts également. La disponibilité d'outils appropriés de gestion de la connaissance et d'un hardware adapté représente certes une facilité pour l'ingénieur de la connaissance, mais fait souvent monter le budget de façon catastrophique.

3.2.2. CONCEPTUALISATION.

Cette phase a pour objectif de décrire complètement les concepts-clés et interrelations dégagés lors de l'étape d'identification du problème, et d'examiner les concepts et relations secondaires qui entrent également dans la base de connaissances du système. En d'autres termes, il s'agit ici de préciser le schéma conceptuel de la base, avec tous ses concepts et leurs interrelations.

L'ingénieur de la connaissance peut trouver utile de tracer des diagrammes représentant la structure de la base de connaissance, et la structure des étapes du raisonnement de l'expert. Ces diagrammes serviront de base à l'étape de formalisation et à l'implémentation de la maquette. C'est donc à ce niveau que l'ingénieur peut commencer à décrire l'architecture globale du système expert et de la base de connaissances, et les mécanismes de contrôle qui viendront se greffer sur le raisonnement de l'expert. Ces mécanismes servent en général à abrégier le temps de travail du système.

La phase de conceptualisation est un processus interactif entre l'expert du domaine et l'ingénieur de la connaissance, et les questions suivantes devraient y être abordées :

- quels types de données sont disponibles ?
- qu'est-ce qui est donné, et qu'est-ce qui est déduit ?
- les sous-tâches ont-elles un nom ?
- les stratégies ont-elles un nom ?
- y a-t-il des hypothèses partiellement identifiables qui sont généralement utilisées ? quelles sont-elles ?

- quels sont les objets apparentés dans le domaine ?
- peut-on faire un diagramme de la hiérarchie des tâches et identifier des relations telles que des relations causales, des inclusions, ou d'autres relations bien définies ?
- quels sont les traitements qui interviennent dans la résolution du problème ?
- quelles sont les contraintes sur ces traitements ?
- quel est le flux d'information ?
- peut-on séparer la connaissance nécessaire pour résoudre un problème de celle utilisée pour justifier la solution ?

3.2.3. FORMALISATION.

La formalisation est un processus qui consiste à faire entrer les concepts-clés, les relations et les sous-problèmes dégagés aux deux phases précédentes dans des représentations plus formelles. C'est lors de cette phase que l'ingénieur de la connaissance joue son rôle le plus actif.

L'ingénieur de la connaissance doit s'efforcer de trouver un ou des modèles susceptibles de représenter le réel perçu et les concepts dégagés lors de la phase précédente. Pour ce faire, il dispose de son expérience dans la résolution de cas similaires et d'ouvrages rédigés sur les différents modèles habituellement utilisés : modèles mathématiques, statistiques, analytiques, etc. Il va essayer de dégager celui qui se rapproche le plus de la structure des concepts et du traitement du problème. Le rapprochement se marque par les modifications qu'on doit apporter à la structure des concepts pour les faire entrer dans le formalisme choisi. Par exemple, un concept décrit par un ensemble de caractéristiques "entrera" mieux dans un élément structuré du modèle que dans une suite d'éléments simples. On choisira alors une modélisation qui permet les objets structurés, plutôt qu'une autre qui ne les permet pas.

Pour choisir la formalisation à adopter, l'ingénieur étudiera la structure de l'espace de recherche des solutions : est-il fini ou non? les concepts sont-ils simples ou structurés? etc. Après détermination de cette structure, l'ingénieur dégagera une modélisation soit mathématique et analytique, soit de comportement, du raisonnement suivi par l'expert pour résoudre le problème posé. Enfin, il veillera à préciser complètement la nature des données et du raisonnement de l'expert du domaine pour déterminer la formalisation qui leur convient le mieux.

Le résultat de la phase de conceptualisation des éléments et des flux dégagés lors de la conceptualisation est une spécification partielle pour la construction de la base de connaissance d'un prototype du système.

Il est possible que l'ingénieur de la connaissance décide de choisir des outils différents, qu'il devra ensuite intégrer les uns aux autres. Ce choix doit être minutieusement examiné, et balancé par la difficulté d'intégrer des techniques différentes. Un exemple : l'ingénieur pourrait penser à des langages différents suivant les parties du système.

En conclusion, on dispose à la fin de cette phase d'une connaissance formalisée et d'outils pour la résolution du problème.

PRESENTATION DE LA DEMARCHE DE HAYES - ROTH

3.2.4. IMPLEMENTATION.

La phase d'implémentation consiste à coder le résultat de la phase précédente en fonction des outils de représentation et de manipulation choisis. L'ingénieur devra veiller à implémenter toute la connaissance préalablement conceptualisée et formalisée.

Le résultat de cette étape est un programme exécutable, prototype du système expert définitif.

3.2.5. TESTS.

La phase de tests consiste à bâtir des jeux de tests, et à faire s'exécuter le prototype sur ces jeux. La construction des jeux se fera suivant des critères tels que le recouvrement de classes de données, dans lesquelles on teste une valeur de la classe et les valeurs limites. Si les jeux de tests sont bien établis, on testera au moins un cas de toutes les classes possibles.

L'ingénieur expérimenté testera plus précisément les trois points suivants, qui révèlent le plus d'erreurs : l'entrée des données et la sortie des résultats, les règles d'inférence et les stratégies de contrôle.

Les caractéristiques principales des entrées et sorties sont l'acquisition de données et la présentation de la conclusion.

L'acquisition de données peut être mauvaise parce que les réponses saisies ne sont pas adéquates, parce que les questions posées ne sont pas pertinentes, ou encore parce que l'information récoltée est insuffisante ou non-cohérente avec la méthode de résolution du problème. Ainsi, les questions peuvent être ambiguës, difficiles à comprendre ou non-adaptées au savoir de l'utilisateur. De plus, le fait de devoir entrer les réponses en tapant sur un clavier peut entraîner pas mal d'erreurs du style faute de frappe non détectée avant le <RETURN>, etc. L'entrée tactile des données, ou la reconnaissance de la voix semblent remporter les suffrages chez les utilisateurs.

Il faudra donc veiller à la complétude de l'acquisition de données, et à sa convivialité.

La sortie des résultats peut également être inadéquate. Il peut y avoir trop ou pas assez de conclusions, avec trop ou pas assez d'hypothèses intermédiaires spécifiées. Les conclusions peuvent être mal organisées, illisibles, trop verbeuses ou trop sommaires. Ici aussi, il faut veiller à la convivialité, et penser de temps en temps qu'un diagramme peut éclaircir mieux que de longues phrases.

PRESENTATION DE LA DEMARCHE DE HAYES - ROTH

L'endroit le plus évident où aller chercher des erreurs de raisonnement est évidemment dans l'ensemble des règles d'inférence du programme. Les règles sont rarement indépendantes entre elles, et il faut vérifier leur cohérence, leur compatibilité, leur correction, et surtout leur présence. L'ingénieur ou le programmeur peut avoir oublié d'introduire une ou plusieurs règles, ce qui fait évidemment que le programme ne tourne pas. Un conséquent ou un antécédent incorrect dans une règle affecteront sa bonne logique, puisqu'elle ne sera pas appliquée dans un environnement prévu. Si les conséquents et antécédents sont corrects, il faut encore veiller à ce que leur accouplement dans la même règle soit bien correct.

Enfin, l'ensemble de règles doit être cohérent, et ce même augmenté des conclusions intermédiaires. Il faudra se méfier des conclusions intermédiaires cohérentes lorsqu'elles sont isolées, et incorrectes lorsqu'elles sont groupées.

Une partie des erreurs d'un système expert se trouve dans ses stratégies de contrôle. Les séquences de consultation de certaines données peuvent changer lors des différentes phases; il faut donc veiller à ce que toutes les données soient consultées. Par exemple, devoir consulter une donnée X avant une donnée Y peut entraîner le fait que la donnée Y ne sera jamais consultée si la donnée Y est placée avant X dans la base. En effet, le marqueur prendra X et ne reviendra pas sur Y sauf si le programme le lui demande expressément.

3.3. DEGAGEMENT DES SPECIFICITES DU KNOWLEDGE ENGINEERING

3.3. DEGAGEMENT DES SPECIFICITES DU KNOWLEDGE ENGINEERING.

Remarque préliminaire : La discussion présentée ci-dessous relève de notre expérience personnelle, sans autre prétention.

La première différence qui apparaît entre le knowledge et le software engineering se marque dans le choix des modèles. En effet, le mode de raisonnement de l'expert du domaine n'est pas aisément modélisable dans les approches classiques telles que celle proposée dans [BOD83] pour la phase d'analyse fonctionnelle. Cette approche implique que l'ordre dans lequel se déroulent les diverses fonctions soit figé, et que les transferts d'information entre les différents modules du système soient toujours définis complètement. En d'autres termes, il est possible dans l'approche classique de tracer les diagrammes de flux d'information entre les différents modules et de définir les caractéristiques de l'information qui circule. Dès lors, l'approche classique convient parfaitement pour des problèmes à résolution algorithmique.

Si le mode de résolution du problème est heuristique, il est impossible de décrire la suite des actions à entreprendre pour parvenir à la solution. Dès lors, on ne peut créer de diagrammes de flux d'information, et il est extrêmement difficile de prévoir les caractéristiques de l'information qui est véhiculée à travers les fonctions.

Les modèles proposés pour la formalisation de problèmes traités par le knowledge engineering sont de type analytique, c'est-à-dire que la situation courante est analysée par le système, et que c'est le résultat de cette analyse qui détermine la tâche qui sera effectuée ensuite.

Le mécanisme d'héritage (ou transmission) des propriétés exige également un formalisme non-classique des concepts. En effet, il faut pouvoir représenter dans la formalisation choisie des relations de généralisation / spécialisation. Nous avons pensé à des arbres sémantiques. L'intelligence artificielle pourvoit un outil intéressant pour représenter ce mécanisme : les frames. En PROLOG, elles ne sont pas implémentées. Il faut donc essayer de s'en tirer autrement, par listes imbriquées, par exemple.

La représentation des connaissances est très déclarative, et les connaissances sont stockées pratiquement en vrac dans la base. Il existe peu de contraintes d'ordonnancement parmi les règles, et la structure de la base de connaissances peut donc consister en une simple suite de faits et de règles. Il ne faut pas jouer avec des outils tels que les SGBD. Le moteur d'inférence du langage employé se charge de marquer les faits et de venir les chercher dans la base par un système de pointeurs absolument transparent à l'informaticien.

3.3. DEGAGEMENT DES SPECIFICITES DU KNOWLEDGE ENGINEERING

Le fait de traiter des informations incomplètes, voire erronées, n'est pas rare dans le domaine de l'intelligence artificielle. Par contre, il l'est plus en programmation classique. Dès lors, le système doit être pourvu de structures de contrôle de haut niveau, susceptibles d'orienter le déroulement du programme sur des bases incomplètes.

4.1. IDENTIFICATION DU PROBLEME A TRAITER

CHAPITRE 4 : CONSTRUCTION DE CONSEIL.

Ce chapitre est découpé de la même façon que le troisième et reprend les étapes cruciales de l'analyse et leurs résultats.

4.1. IDENTIFICATION DU PROBLEME A TRAITER.

4.1.1. IDENTIFICATION DES PARTICIPANTS ET DE LEUR ROLE.

Le public à qui s'adresse cet outil était mal défini au départ, et plusieurs solutions étaient envisagées suivant le type de préoccupations professionnelles de l'interlocuteur. Nous avons dû effectuer un choix parmi ces différents types d'utilisateurs afin de savoir quels experts nous interrogerions. Nous allons présenter les différents destinataires potentiels de ce système, la solution qui a été retenue, et les experts que nous avons sélectionnés ainsi que leur rôle.

4.1.1.1. Les commerciaux.

Une première possibilité est que le public concerné soit le "grand public", c'est-à-dire tout qui va acheter de la peinture chez un droguiste ou dans une grande surface, et désire des conseils d'emploi et de choix de produits. L'outil pourrait prendre la forme d'une cassette à introduire dans un lecteur, et qui prodiguerait des conseils-peinture.

Plusieurs problèmes se posent dans cette perspective. Un premier problème est celui de l'engouement publicitaire. En effet, deux produits chimiquement identiques qui se vendent sous deux marques différentes n'occupent pas forcément la même position dans la hiérarchie des succès commerciaux. En particulier, le même produit d'une marque peut très bien marcher sous la marque principale, et très mal pour une filiale qui désirerait le commercialiser.

Exemple : T.P.C. a repris un produit d'une de ses filiales, et l'a développé sous un autre nom dans son propre éventail de produits. L'opération s'est soldée par un fiasco commercial, alors que le produit était très bien vendu par la filiale.

Un autre problème réside dans le fait que les critères employés par un commercial pour conseiller un produit sont souvent de type commercial et sont peu rationnels. Dès lors, ils sont difficilement formalisables.

Exemple : on n'applique pas de peintures inhibitrices de rouille sur un mur, parce qu'elles sont conçues pour les métaux. En fait, si un utilisateur veut un aspect pailleté sur un mur, il n'a qu'une seule solution : y appliquer une peinture inhibitrice de rouille, ce sont les seules qui présentent un effet de ce genre, et il n'existe aucune contre-indication.

4.1. IDENTIFICATION DU PROBLEME A TRAITER

4.1.1.2. Les vendeurs.

Une deuxième possibilité est que l'outil puisse être utilisé en tant que moyen didactique pour les personnes engagées comme revendeurs, pour les grossistes ou encore pour les peintres professionnels. En effet, l'apprentissage des règles de mise en peinture et des caractéristiques des produits se fait souvent sur le tas. De ce fait, c'est un processus relativement lent, et qui n'est pas sans risque : si la personne qui guide le nouveau conseiller commet des erreurs, son élève commettra les mêmes erreurs, et de même ceux qui seront ses propres élèves. De plus, chaque conseiller a ses produits fétiches qu'il connaît parfaitement et une connaissance moins approfondie d'autres produits de la gamme, qui pourrait induire des erreurs dans le conseil. La disposition d'un outil d'apprentissage exhaustif permettrait d'accélérer le processus, et d'éviter les erreurs dues à de telles lacunes dans la connaissance.

De ce point de vue, l'outil serait basé sur des notions techniques, pour ne pas dire chimiques. Dès lors, il apporterait toutes les solutions techniquement possibles pour résoudre un problème et sa réponse ne sera pas biaisée par l'aspect marketing.

4.1.1.3. Les chimistes.

L'outil serait utilisé à des fins de recherche et devrait aider à repérer des utilisations non-conventionnelles des produits, du type de celle décrite plus haut, mais pourtant valables techniquement. De même, le repérage des cas pour lesquels il n'existe pas de système de peinture pourrait être plus aisé, et pourrait donner lieu à la mise au point de nouveaux produits pour lesquels il existerait des débouchés. L'outil pourrait également servir à vérifier la cohérence de gammes de produits (pas de double emploi, pas de trous, ...).

Il est évident que dans ce cas, les critères qui déterminent un système de peinture sont rationnels, puisque basés sur la composition chimique des produits, et donc aisément formalisables.

Le seul problème dans ce cas est l'étroitesse du public par rapport à la masse des connaissances à traiter. Il faudrait également descendre au niveau des molécules du produit pour déterminer la solution, et cette approche nous paraît être un peu trop spécialisée par rapport à ce qui nous était demandé, c'est-à-dire qu'elle requiert trop de connaissances techniques du domaine.

4.1.1.4. Point de vue adopté.

A ce moment de l'analyse, le public-cible du système conseil n'était donc pas précisé et nous nous sommes rendu compte que quel qu'il soit, il était indispensable de disposer de critères techniques formalisables. Ainsi, il nous semble que notre système pourrait prendre la forme d'un noyau technique qui serait un prototype, autour duquel, à l'occasion de développements ultérieurs, viendraient se

4.1. IDENTIFICATION DU PROBLEME A TRAITER

greffer des couches spécifiques à l'utilisation par l'un ou l'autre domaine.

Dans un premier temps, le prototype donnera comme résultat un ou plusieurs système(s) de peinture techniquement correct(s), mais ne tenant pas compte de critères commerciaux. Bien entendu, ces critères (par exemple, le prix) pourront être introduits par la suite sous forme d'échelles de valeurs.

La solution adoptée est donc de forme hybride. Les connaissances ont été recueillies en grande partie chez les chimistes, tout en restant au niveau des familles de composants, et nous avons également discuté avec les informaticiens du service.

Leur rôle reste cependant assez limité et consultatif : les entrevues que nous avons eues avec les experts nous ont permis de discuter de points ayant à voir avec leur domaine uniquement, et nous ne leur avons jamais exposé les problèmes informatiques auxquels nous devons faire face. CONSEIL a donc été élaboré par des informaticiens sur base de renseignements fournis par des experts chimistes et n'est pas le fruit d'une collaboration constante.

Le but de nos discussions avec les informaticiens était de ne pas déranger continuellement les experts chimistes pour obtenir des renseignements d'ordre général sur la peinture. En effet, ils nous ont donné un certain nombre d'éclaircissements pour ce qui concernait la compréhension de termes couramment utilisés dans le domaine, et pas toujours purement chimiques.

4.1.2. IDENTIFICATION DE L'OBJECTIF.

L'objectif de ce travail est de montrer qu'il est possible de construire un outil informatisé dédié au conseil en peinture. Le conseil donné doit être correct et cohérent par rapport aux connaissances introduites dans la base et le mécanisme de raisonnement du système ne peut pas trahir celui d'un expert qualifié pour ce travail. En d'autres termes, les solutions préconisées par le système automatique doivent être les mêmes que celles données par l'expert humain et la façon de les obtenir doit être la plus proche possible de la démarche humaine.

Le système doit être utilisable par des personnes qui voudraient parfaire leur expertise en ce sens qu'il donne toutes les solutions techniquement correctes à un problème, et ne commet pas d'erreurs de jugement par rapport au contenu de sa base de connaissances.

Parallèlement, ce projet permet de montrer qu'il est possible de formaliser le travail d'un expert, à condition que ses décisions soient logiquement motivées et que les informations dont le système dispose soient rigoureusement définies et conformes avec la réalité.

Enfin, un objectif annexe est d'évaluer l'intérêt et l'applicabilité de PROLOG comme langage de représentation de

4.1. IDENTIFICATION DU PROBLEME A TRAITER

connaissances et comme moteur d'inférence pour l'implémentation de systèmes experts "en vraie grandeur".

4.1.3. PREMIERE IDENTIFICATION DU PROBLEME.

Il s'agit de construire un système expert de conseil de mise en peinture, capable de donner la séquence des traitements à effectuer sur un objet préalablement décrit pour l'amener dans un état déterminé par l'utilisateur, et caractérisé par un ensemble de critères. Ces critères peuvent être, par exemple, le nom de l'objet, sa matière, l'environnement dans lequel il est placé, l'aspect qu'il doit présenter une fois traité, les garanties à fournir sur l'état final, etc.

La construction classique d'un système de peinture est exposée en annexe A.

Le système doit donner toutes les solutions possibles et techniquement réalisables au problème posé. Dès lors, il fait abstraction des stratégies commerciales de la firme.

Il doit pouvoir expliquer le raisonnement qu'il a suivi pour aboutir à chaque solution qu'il préconisera, et donc être pourvu d'une interface explicative.

Il doit être facilement utilisable par un non-informaticien, et donc il faut lui adjoindre une interface de façon à ce que les communications système - utilisateur puissent se dérouler dans un sous-ensemble du langage naturel.

Enfin, il doit être aisément modifiable, et adaptable aux modifications induites par la politique des produits de la firme, telles que ajout d'un produit, suppression d'un produit, changements dans la composition chimique d'un produit qui pourraient induire de nouvelles applications ou en supprimer d'autres, etc.

Les interfaces d'explication, de dialogue et d'extension n'entrent pas dans le cadre de ce travail; elles font l'objet du mémoire d'Isabelle WILMET et Yvan VOTOT, à paraître en 1987.

Après cette présentation informelle basée sur la définition de ce que doit être un système expert, nous allons tenter de répondre aux questions suggérées au chapitre 3.

Le problème peut être vu comme un problème de diagnostic. Si on reprend le tableau comparatif établi au point 2.1.2. entre deux systèmes experts de diagnostic, on peut en dériver le tableau de la page suivante, qui établit la correspondance entre les principes généraux relatifs aux systèmes experts et CONSEIL.

4.1. IDENTIFICATION DU PROBLEME A TRAITER

DOMAINE	CONSEIL EN PEINTURE
SUJET	support à peindre
ETAT INITIAL =SYMPTOMES	état du support avant tout traitement, reprenant ses dégradations, ses traitements antérieurs, etc
ETAT FINAL	état du support après application de la séquence de traitements, résultat du système
DIAGNOSTIC	différence entre les états initial et final
REMEDE	séquence résultat de traitements

Le système CONSEIL est destiné à résoudre un seul type de problème : à partir de deux états, l'un initial et l'autre final, d'un support à peindre, donner la séquence des traitements que le support dans son état initial doit subir pour atteindre son état final. La séquence de traitements vide est une solution acceptable.

Les états terminaux doivent être décrits selon des critères communs de façon à pouvoir les comparer aisément. La séquence résultat devrait être documentée de renseignements sur les traitements conseillés tels que leurs modalités d'application, le matériel à utiliser pour leur application, les précautions à prendre, etc.

Le problème peut se décomposer en trois sous-problèmes principaux que nous subdiviserons à leur tour, de façon à établir une hiérarchie des tâches.

Le premier sous-problème est l'introduction d'une description des états initial et final par l'utilisateur. Il s'agit ici de faire connaître au système les caractéristiques des deux états, dont le premier représente le support au moment de la requête, avec ses dégradations et ses traitements antérieurs, et le second le support tel qu'il sera lorsque la séquence résultat lui aura été appliquée. Dans

4.1. IDENTIFICATION DU PROBLEME A TRAITER

la description des traitements antérieurs, seuls certains points sont importants : la composition chimique du dernier traitement pour des raisons de compatibilité, et les caractéristiques que les traitements antérieurs ont conférées au support, du type résistance à certains éléments, protection contre des dégradations futures, etc.

Le deuxième sous-problème est la construction de la séquence résultat à partir de ces deux états. Ceci peut être formulé de la façon suivante:

Etant donné une séquence de traitements partiellement construite, menant à l'état final désiré, à partir d'un état intermédiaire lui correspondant, deux types de problèmes se posent :

(1) il faut d'abord déterminer un nouveau traitement permettant d'atteindre un nouvel état intermédiaire. Ce choix se fait à l'issue de deux analyses :

(1.1) une comparaison entre l'état intermédiaire "cible" et l'état du support après application d'un traitement candidat. Les caractéristiques de ce dernier doivent être compatibles avec l'état "cible", c'est-à-dire que l'état "cible" doit être considéré comme une occurrence particulière de l'état du support après application du traitement.

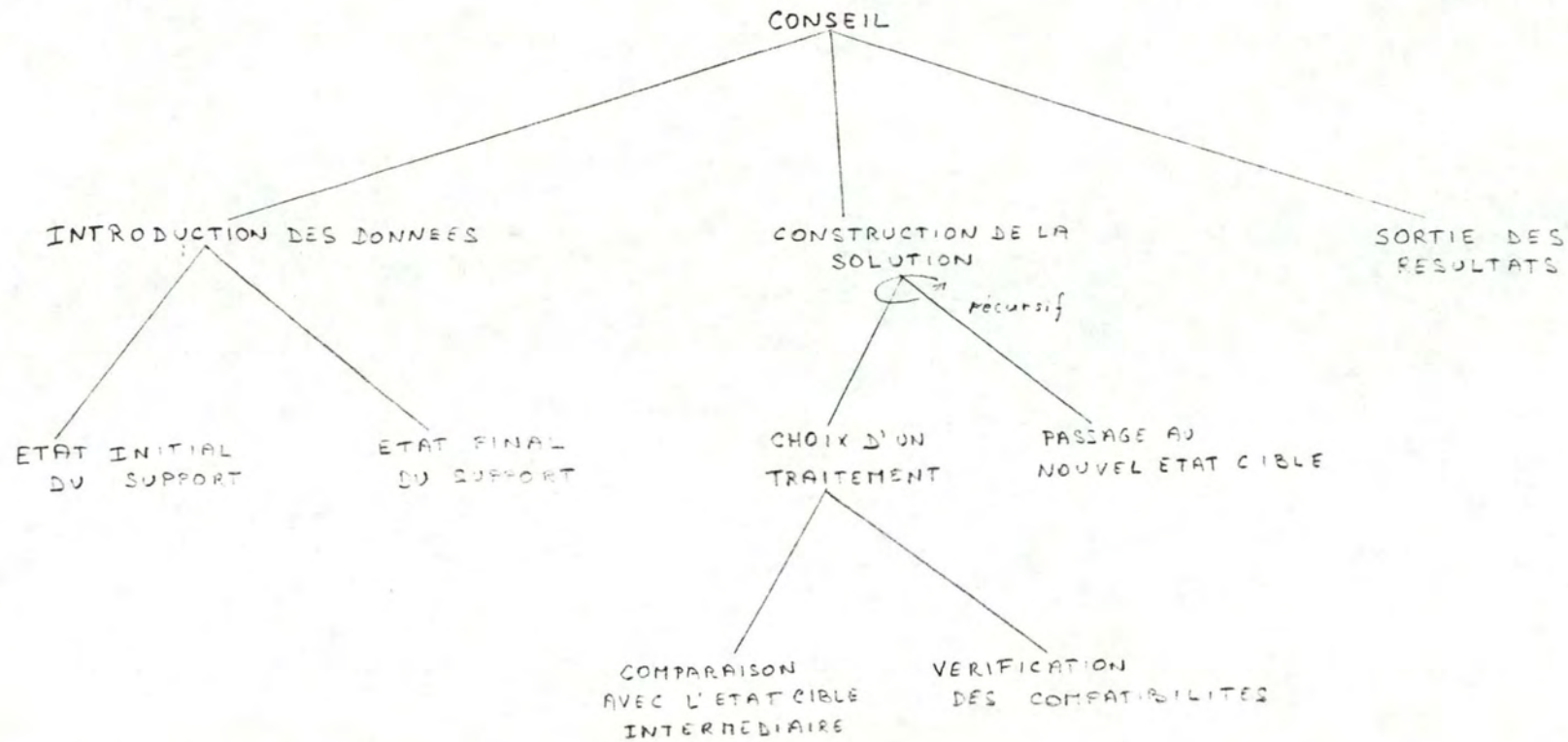
(1.2) une vérification de la compatibilité chimique entre un traitement candidat ayant franchi l'étape (1.1) avec succès et le dernier traitement choisi dans la séquence construite jusqu'alors.

(2) une fois déterminé un traitement candidat d'après les étapes décrites en (1), il faut mettre à jour l'état intermédiaire comme étant le nouvel état "cible".

Ce processus itératif se termine lorsque le nouvel état intermédiaire obtenu n'est rien d'autre que l'état initial.

Le troisième et dernier sous-problème est la sortie à l'écran de la séquence résultat dérivée par le deuxième sous-problème, ainsi que de la documentation s'y rapportant.

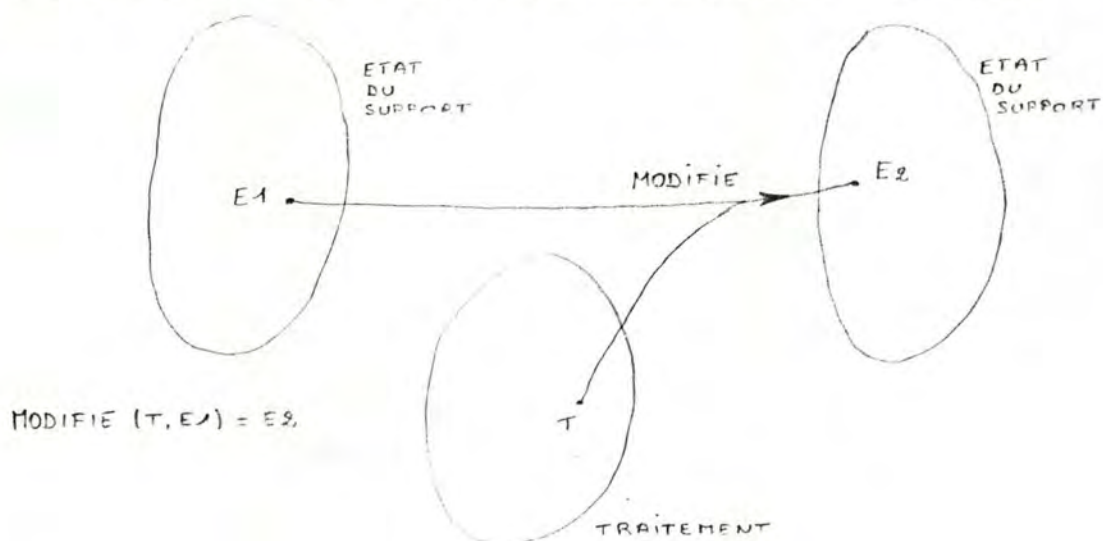
Ainsi, un premier arbre-ET des tâches peut s'établir comme schématisé à la page suivante. Cet arbre est une étude détaillée du processus de construction de la séquence résultat sous forme d'une approche top-down du problème. Ainsi, une tâche (un noeud de l'arbre) est subdivisée en sous-tâches (les fils de ce noeud). Les tâches qui constituent les feuilles de l'arbre sont considérées comme élémentaires à ce stade du développement.



4.1. IDENTIFICATION DU PROBLEME A TRAITER

Les seules données du problème sont donc les descriptions des états initial et final.

Examinons à présent les objets importants du problème et leurs interrelations. Un premier concept est celui d'ETAT DU SUPPORT, qui est défini comme l'état d'un objet susceptible de subir un traitement. Le second est celui du TRAITEMENT, qui intervient comme élément dans une séquence résultat, et est défini soit comme l'application d'un produit de la gamme TRIMETAL soit comme une opération de préparation. La relation qui lie ces objets est la relation MODIFIE, qui exprime que l'application d'un TRAITEMENT sur un SUPPORT dans un ETAT donné modifie cet ETAT. Cette relation peut se schématiser comme suit.



4.1.4. NATURE DE L'EXPERTISE REQUISE.

Dans la résolution de ce genre de problème, l'expertise humaine joue un grand rôle, et sous plusieurs aspects. Les experts plus jeunes emploient des connaissances chimiques pour conseiller une suite de traitements résultat, ou se basent sur les systèmes de peinture conventionnels de la firme, qui eux-mêmes ont été élaborés par des experts. Les plus âgés possèdent une expérience dans le traitement de cas conventionnels et particuliers. Cette expérience peut être entretenue par un feed-back provenant des clients qui ont été conseillés. Le plus souvent, chacun a ses solutions "préférées", mais le conseil est toujours établi dans un esprit de faisabilité et de validité et dans un désir de satisfaire au mieux les exigences diverses du client.

D'autres aspects que la chimie peuvent entrer en ligne de compte, aspects que nous n'abordons pas dans CONSEIL. Nous pensons en particulier à l'aspect financier de la solution : le client peut dicter des exigences de coût que l'expert se doit de respecter. Ce sont des caractéristiques intéressantes mais difficiles à traiter dans la première version du projet.

En outre, les experts disposent d'une documentation complète sur les produits établie par les laboratoires, et de fiches techniques

4.1. IDENTIFICATION DU PROBLEME A TRAITER

destinées plus spécialement à l'usage des peintres, et décrivant les modalités d'application des traitements. On trouvera, en annexe B, un exemple d'une telle fiche.

Le savoir-faire de l'expert se mesure à sa capacité de pouvoir agencer tous ces renseignements de façon à fournir un conseil correct et cohérent par rapport aux exigences du client. De plus, il doit donner son avis assez rapidement et être capable de traiter des cas très particuliers en convainquant le client du bien fondé du conseil donné. L'expert doit enfin être capable de discuter avec le client qui a des exigences incompatibles avec la gamme de produits, de façon à lui conseiller la solution qui se rapproche le plus de ses exigences.

Ainsi, l'étendue de l'expertise est énorme, et répartie sous différents aspects : la connaissance chimique des produits et leurs (in)compatibilités, la connaissance de cas particuliers qui réclament une utilisation non-conventionnelle et parfois paradoxale d'un produit (par exemple application d'un primer anti-corrosion sur un mur pour obtenir un effet pailleté), la connaissance des aspects financiers des produits, un minimum de diplomatie et un maximum de conviction dans les relations avec le client, et la rapidité d'intervention.

Certaines situations sont cependant insolubles, et ce, pour plusieurs raisons. Le produit dévolu à l'usage particulier que le client veut en faire ne se trouve pas dans la gamme. Cette situation se présente rarement dans le cadre de la peinture en bâtiments. Le support peut être trop abimé pour être traité : un béton très fissuré ou un métal rongé de rouille sont intraitables à long terme. Les traitements appliqués au support antérieurement peuvent être inconnus. Dès lors, l'expert ne peut conseiller un système qui "marche à coup sûr", et ne peut que préconiser de décaper complètement l'objet ou de faire des essais avec des produits différents.

De tels obstacles déclencheront dans le système conseil l'occurrence de la solution "pas de solution" quand l'état du support décrit initialement ne peut être corrigé ou que le problème ne peut être résolu grâce à un produit de la gamme TRIMETAL, ou l'occurrence d'une solution dont le premier traitement est le décapage s'il y a une incompatibilité entre les traitements antérieurs subis par le support et les traitements préconisés par le système conseil. Cette dernière possibilité constitue un premier pas vers des extensions futures du système qui deviendrait réellement expert dès lors qu'il pourrait "raisonner" en univers incomplet ou par incertain (méconnaissance des traitements déjà subis).

4.1. IDENTIFICATION DU PROBLEME A TRAITER

4.1.5. IDENTIFICATION DES RESSOURCES.

Nous avons consulté deux types de ressources, celles concernant plutôt le domaine d'expertise, et celles concernant les techniques d'implémentation envisageables pour mener notre entreprise à son terme.

En ce qui concerne le domaine d'expertise, nous nous sommes familiarisées avec la chimie des peintures, la philosophie des systèmes de peinture et les spécificités des peintures au moyen de documents écrits. Les experts ont comblé les lacunes de ces documents, et nous ont fait saisir ce que signifiait "donner un conseil en peinture", et quels sont les mécanismes mis en oeuvre pour cette tâche. On trouvera en annexe les résultats de certaines de ces conversations.

En ce qui concerne les techniques d'implémentation, nous avons eu recours à des ouvrages traitant d'Intelligence Artificielle en général, et de systèmes experts en particulier, et à des ouvrages expliquant les mécanismes de la programmation logique et de PROLOG.

Nous disposions de quatre mois pour obtenir les renseignements utiles à la construction de CONSEIL et de quelques heures de disponibilité des experts. Ce fut malheureusement bien peu.

Pour ce qui est des moyens financiers et matériels, nous disposions d'un ordinateur HP-150, doté d'un Winchester d'une capacité de 10 Mbytes, et d'une imprimante. Le langage de programmation du prototype est le PROLOG d'Edimbourg, version 1, dépourvu d'un environnement de programmation.

4.2. CONCEPTUALISATION

4.2. CONCEPTUALISATION.

Il s'agit toujours d'un processus interactif entre les experts et nous-mêmes. Nous avons travaillé sur un résultat partiel obtenu au cours de la première phase pour pouvoir confronter plus rapidement notre conceptualisation à l'activité réelle du conseil-peinture. Nous avons choisi comme noyau un sous-système constitué des supports de type métallique.

4.2.1. CHOIX D'UN SOUS-SYSTEME.

Nous avons pris la décision de nous restreindre dans un premier temps au sous-système constitué par les supports de type métallique. La raison pour laquelle nous avons décidé de nous limiter à un sous-système réside dans le fait qu'un système complet constitue un ensemble énorme de traitements et de supports, et qu'il nous a semblé pertinent, dans un premier temps, de limiter le problème sans s'encombrer d'un volume trop important de données. L'important du projet est d'automatiser le principe de construction d'un système de peinture élaboré par un expert humain.

La raison pour laquelle nous avons choisi le sous-ensemble constitué par les supports de type métallique provient du fait que c'est pour cet ensemble de supports que la documentation est la plus rigoureuse, la plus facilement disponible et la plus complète.

La démarche suivie pour les supports métalliques pourra être facilement étendue à des supports d'autres types par la suite. En effet, cette démarche est la même quel que soit le type de support considéré. En particulier, les concepts dégagés au cours de cette phase restent identiques de même que leurs caractéristiques. Evidemment, les valeurs possibles pour chacune de celles-ci sont sujettes à modification et devront être complétées lors d'extensions futures du système conseil.

Considérer un sous-ensemble de supports implique une réduction du nombre de traitements applicables, vu que certains traitements sont propres à un type de support. Par exemple, le bois ne doit jamais être dérouillé, et le métal n'est jamais envahi d'insectes xylophages, pas plus que les maçonneries.

Remarquons enfin que le sous-système ne comprend que des produits d'utilité courante, et ne traite donc pas des produits conçus pour un usage industriel.

4.2. CONCEPTUALISATION

4.2.2. LES CONCEPTS.

Au fil de nos discussions avec les experts au sujet de ce sous-système, sont apparus des éléments tels que la composition chimique d'un traitement qui détermine sa résistance, les protections qu'il assure, son adhérence sur un support (que ce soit un matériau nu ou un support ayant déjà fait l'objet d'un traitement), etc.

Pour structurer ces diverses informations, nous nous sommes posé la question suivante : "Que recherche une personne qui demande un conseil-peinture?".

Un client recherche essentiellement un effet esthétique auquel s'ajoutent des exigences telles que la remédiation à certaines détériorations du support à peindre ou bien la résistance à des facteurs particuliers. Outre ces objectifs explicites, il faut tenir compte de la nature du matériau utilisé pour la fabrication de l'objet à peindre, des contraintes du milieu dans lequel il se trouve et de l'état dans lequel il est.

Parallèlement, il fallait trouver les caractéristiques du traitement de façon à exprimer le plus simplement possible la relation "modifie". En d'autres mots, il s'agissait, pour chaque caractéristique du support, d'associer un facteur particulier du traitement susceptible de la modifier.

Examinons de plus près les deux concepts primordiaux dégagés lors de la première phase : les supports et les traitements.

4.2.2.1. Les supports.

Nous avons adopté le terme SUPPORT pour tout objet susceptible d'être peint. En effet, peu importe le nom de l'objet à traiter, pourvu que l'on connaisse ses caractéristiques qui peuvent influencer les traitements que l'on peut lui appliquer.

Ces caractéristiques sont au nombre de neuf :

- sa nature,
- sa forme,
- son environnement,
- son état de surface,
- son aspect,
- ses résistances,
- ses protections,
- les traitements qui lui ont déjà été appliqués,
- ses caractéristiques spéciales.

Ces différentes caractéristiques peuvent être définies de la façon suivante :

- nature : type de matériau utilisé pour la fabrication du support.
Par exemple : acier, galvanisé, cuivre.

4.2. CONCEPTUALISATION

- forme : situation dans l'espace de la surface à peindre ou forme générale de l'objet.
Par exemple : horizontale ou non, angulaire, tubulaire.
- environnement : conditions dépendant du milieu environnant le support (entre autres atmosphériques et chimiques).
Par exemple : marin, rural, intérieur agressif.
- état de surface : appréciation de l'état du support à peindre par rapport à sa recouvrabilité.
Par exemple : corrodé, recouvert d'un produit en bon état, sale, non-poussiéreux.
- aspect : type d'aspect au sens utilisé en peinture.
Par exemple : brillant, filmogène, lasure.
- résistances : facteurs agressifs auxquels le support résiste.
Par exemple : une ambiance agressive, chaleur.
- protections : dégradations auxquelles le support reste insensible.
Par exemple : la corrosion.
- traitements déjà appliqués : si le support n'est pas nu, liste des traitements que le support a déjà subis.
Par exemple : Plomoferrine N, Trisatin, ponçage.
- caractéristiques spéciales : facteurs exceptionnels dont il faut tenir compte, mais qui ne rentrent dans aucune autre catégorie.
Par exemple : antidérapant, non-toxique.

4.2.2.2. Les traitements.

Le concept de TRAITEMENT a été retenu pour représenter l'application d'un produit ou une opération de préparation de support. Nous avons pensé qu'une classification des produits basée sur les composants chimiques permettrait de dégager des familles de produits qui possèdent ou non une caractéristique. Par exemple, nous aurions souhaité pouvoir tirer des conclusions telles que "tous les produits dont le solvant est un alkyde adhérent sur le bois". A ce stade, il semble que de telles règles ne puissent être formulées principalement parce que ce travail nécessiterait, de la part des chimistes, plus d'investissements qu'il ne leur est possible de nous consacrer. Cependant, elles ne nous paraissent pas pour autant impossibles à élaborer. Il faudrait de toute façon parvenir à un compromis entre une énumération de chaque caractéristique d'un traitement et des règles basées sur une composition chimique plus complexe. Dans la construction du prototype, nous avons jugé préférable, pour une question de délai et de disponibilité des experts, d'opter pour l'énumération de valeurs des caractéristiques que nous avons identifiées pour tout traitement. Ces caractéristiques sont les suivantes :

4.2. CONCEPTUALISATION

- sa composition chimique,
- ses conditions d'application,
- son aspect,
- ses résistances,
- ses protections,
- ses caractéristiques spéciales.

Ces différentes caractéristiques peuvent être définies de la façon suivante :

- composition chimique : composants chimiques utilisés lors de la fabrication d'un produit, et renseignements dépendant de la composition.
Par exemple : solvant, PVC.
- conditions d'application : conditions matérielles (c'est-à-dire outillage) et atmosphériques à remplir pour l'application du traitement; il s'agit de renseignements portant sur la façon d'appliquer le traitement et d'exigences portant sur le support avant application du traitement.
Par exemple : brosse, T° ambiante, support adhérent.
- résistances : facteurs agressifs auxquels le traitement résiste.
Par exemple : atmosphère industrielle, solvants.
- protections : dégradations contre lesquelles le traitement protège le support.
Par exemple : corrosion, infiltration d'eau.
- caractéristiques spéciales : qualités que le traitement apporte au support et qui n'entrent dans aucune autre catégorie.
Par exemple : antidérapant, non-toxique.

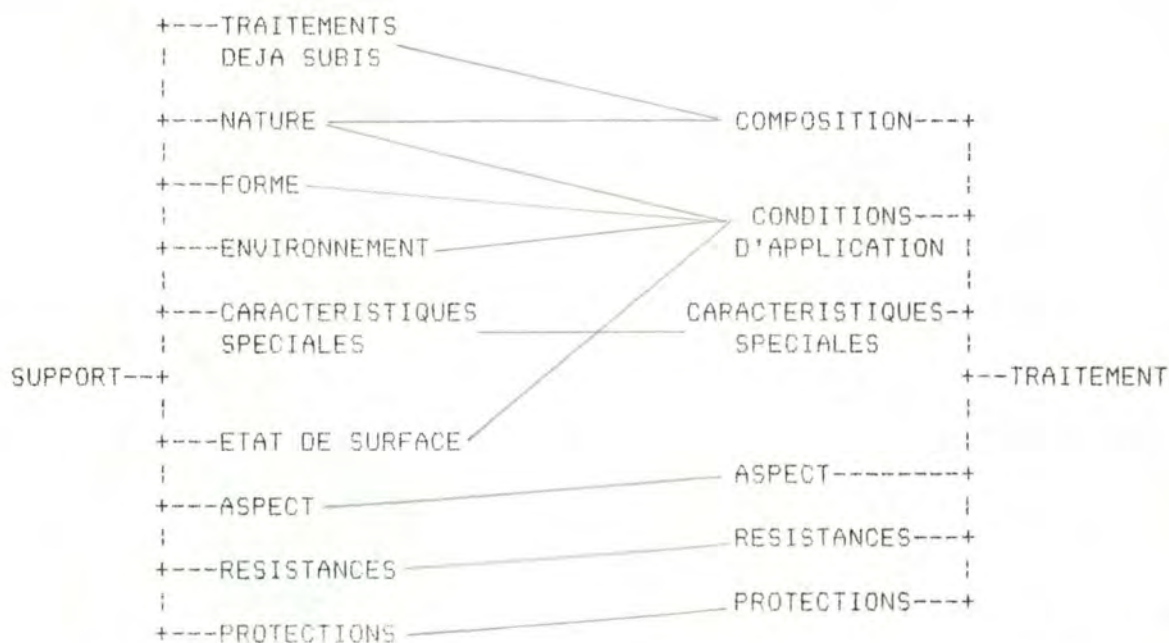
4.2.2.3. La relation MODIFIE.

La conceptualisation ci-dessus permet d'établir entre les caractéristiques des supports et celles des traitements à leur appliquer des relations relativement aisées à exprimer. En effet, il y a correspondance entre les dégradations du support (c'est-à-dire son état de surface) d'une part et les conditions d'application du traitement d'autre part; entre environnement et exigences de résistance du support après finition d'une part, et la résistance du produit à l'extérieur de la couche et la protection préventive assurée au support d'autre part; entre le type du support d'une part et la compatibilité d'un traitement sur ce support d'autre part; et enfin entre l'aspect désiré par l'utilisateur d'une part et l'aspect offert par le traitement d'autre part. De plus, de la description du support, on peut déduire une série de traitements. Par exemple, si le support à traiter est métallique ferreux, il faut prévoir un traitement anti-rouille, même si l'utilisateur ne l'a pas demandé explicitement. S'il est rouillé, le système doit prévoir un grattage ou un brossage de la couche de rouille.

4.2. CONCEPTUALISATION

Rappelons que la relation "modifie" est définie de l'ensemble des supports vers l'ensemble des traitements. Elle est définie ssi les conditions d'application d'un traitement, T, ne sont pas incompatibles avec les valeurs des caractéristiques nature, forme, environnement, état de surface et traitements déjà subis d'un support S. T "modifie" S signifie que l'application de T sur S a pour effet d'attribuer aux caractéristiques aspect, protection, résistance et caractéristiques spéciales de S les valeurs des caractéristiques correspondantes de T. La notion d'incompatibilité sera précisée dans la suite et de façon complète au niveau de la formalisation.

Les correspondances sus-mentionnées entre les caractéristiques des concepts seront établies par le système de la façon schématisée ci-après :



En résumé, les interactions entre les deux concepts retenus sont représentées par les arcs qui décrivent des occurrences potentielles de la relation "modifie"; les occurrences réelles de la relation "modifie" sont régies par les règles de construction d'un système de peinture précisées plus loin.

Remarque :

Notons que dans une version plus élaborée du système, d'autres concepts, d'un niveau supérieur à ceux de support et de traitement, pourraient être définis. Nous pensons par exemple à un concept OBJET qui représenterait un objet réel à peindre, tel une balançoire ou un mur de pierre. Il a une réalité propre, et un nom. Ses caractéristiques peuvent être décrites en termes des neuf caractéristiques du support, plus un nom d'objet.

4.2. CONCEPTUALISATION

4.2.3. ELABORATION D'UN DIAGRAMME DES TACHES.

L'arbre-ET des tâches, esquissé à la section 4.1.3., peut à présent être affiné davantage compte tenu des caractéristiques des états du support et des traitements que nous avons dégagés.

Certains renseignements concernant un traitement, tels que certaines de ses conditions d'application, ne sont d'aucun intérêt lors de la construction de la solution. Ils seront utilisés lors de la mise sur pied de l'interface de sortie des résultats, pour donner à l'utilisateur des informations supplémentaires sur les traitements à appliquer.

Nous allons à présent détailler chacune des feuilles de l'arbre présenté à la page suivante.

Lors de l'entrée de l'état initial, l'utilisateur décrira la nature du support, sa forme, son environnement, les protections dont il dispose, son état de surface et les traitements qu'il a déjà subis. Les trois premières caractéristiques sont invariables dans le système, car elles déterminent l'environnement spatial du support.

Si le support a déjà subi des traitements, il est vraisemblable que ceux-ci lui ont assuré certaines protections. Il est utile de les mentionner, car ces exigences ne devront plus être satisfaites par la suite. En effet, ce sont les couches les plus proches du support qui lui assurent ses protections.

Par contre, il est inutile de mentionner les résistances du support : en effet, les résistances doivent être couvertes par la dernière couche du système, à savoir la finition. Dès lors, il importe peu de savoir à quoi résiste le support dans son état initial, puisque toutes les exigences de résistance resteront à satisfaire par la finition.

L'état de surface détermine les dégradations éventuelles du support, et par là la nécessité de lui appliquer des traitements de préparation avant de commencer à le peindre.

La liste des traitements déjà subis servira à assurer la compatibilité du support avec la suite de traitements construite par le système.

L'état final du support présente les exigences de l'utilisateur. La forme, la nature et l'environnement ne sont pas demandés, puisqu'ils sont invariables dans le système et ont été saisis lors de l'entrée de l'état initial.

Les caractéristiques spéciales, les résistances, les protections et l'aspect du support ne peuvent être dérivés de l'état initial. Dès lors, il faut préciser ces exigences, qui guideront la construction de la solution.

La comparaison avec l'état intermédiaire cible se fait de la façon suivante. On compare respectivement :

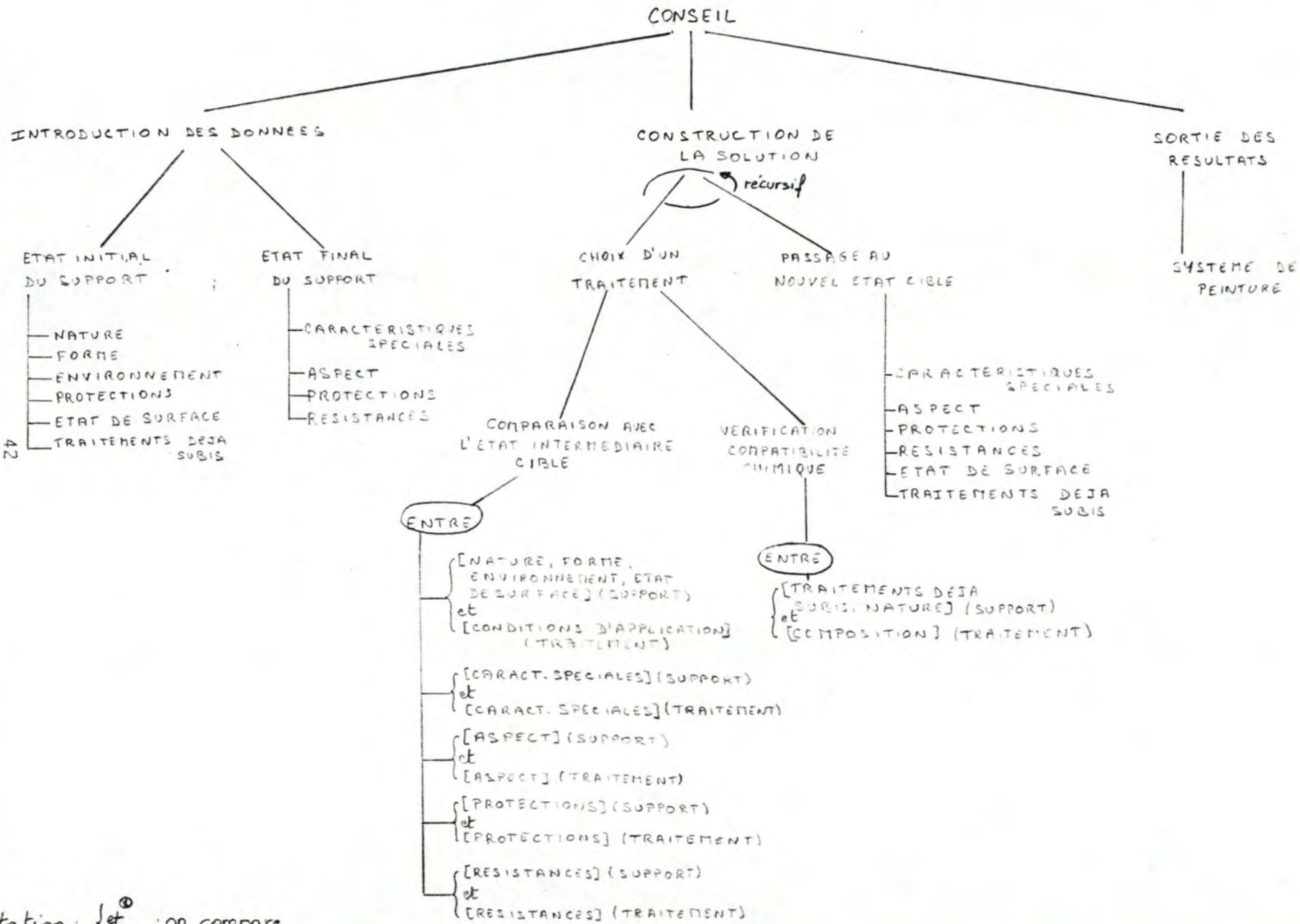
- la nature, la forme, l'environnement et l'état de surface du support avec les conditions d'application du traitement;

4.2. CONCEPTUALISATION

- les caractéristiques spéciales exigées pour le support avec celles du traitement;
- l'aspect exigé pour le support avec celui du traitement;
- et les résistances exigées pour le support avec celles apportées par le traitement.

La compatibilité chimique sera établie par une comparaison des traitements déjà subis et de la nature du support avec la composition du traitement choisi à l'étape précédente.

Lors de l'avancement vers l'état initial, seules les caractéristiques variables du support seront mises à jour, ce qui est évident : un support ne change pas d'environnement spatial lorsqu'on le peint.



4.2. CONCEPTUALISATION

notation : $\left\{ \begin{matrix} \text{et} \\ \text{et} \end{matrix} \right.$: on compare
 ① et ②

4.2. CONCEPTUALISATION

4.2.4. PREMIERE EBAUCHE DES REGLES CONTRAIGNANT LA CONSTRUCTION D'UN SYSTEME DE PEINTURE.

Ce point reprend les règles, pour la plupart de compatibilité chimique, à respecter lors de la construction de la séquence résultat. Elles proviennent de discussions que nous avons eues avec les experts chimistes, et constituent un ensemble non-exhaustif de contraintes d'intégrité à respecter par le système expert. On pourra trouver en annexe des tableaux de compatibilité entre liants qui sont trop longs que pour être repris ici.

Le principe de classification est le suivant : nous énoncerons d'abord les règles dépendant de la compatibilité chimique, classées selon le liant, le solvant, les pigments et des principes généraux concernant l'entièreté du système de peinture. Ensuite, nous citerons les règles qui concernent les composants de l'état du support.

4.2.4.1. Règles dépendant de la composition chimique.

a) liant.

1. Les produits thermoplastiques sont exclus d'un système de peinture lorsqu'ils doivent être soumis à la chaleur.
2. Les alkydes, les époxyds et les polyuréthanes bi-composants sont thermodurcissables.
3. Dans le sens finition --> support, la thermoplasticité de chaque couche doit être décroissante.
4. Les produits thermodurcissables (qui durcissent par oxydation au cours du temps) sont à déconseiller à l'extérieur (parce que l'évaporation y est plus forte).
5. Un produit peu ou pas thermoplastique ne peut s'appliquer sur un support thermoplastique.
6. L'association polyvinylbutyral - acide phosphorique modifie l'état de surface du support au point de vue chimique.
7. Les polyuréthanes mono-composants doivent être appliqués sur un support sec (ils durcissent à l'humidité avec un dégagement de CO₂ => si le support est humide, le film durcit des deux côtés en emprisonnant des bulles de CO₂).
8. Les polyuréthanes bi-composants peuvent être appliqués sur un support légèrement humide.
9. Les polyuréthanes aromatiques sont sensibles aux UV.
10. Les polyuréthanes aliphatiques sont beaucoup moins sensibles aux UV.
11. La sensibilité aux UV donne lieu au jaunissement et au farinage.

4.2. CONCEPTUALISATION

12. Si un produit est très sensible aux UV, alors il ne faut pas le conseiller pour l'extérieur.
13. La famille des époxyds est sensible aux UV.
14. La famille des époxyds est caractérisée par une bonne adhésion sur beaucoup de supports.
15. Les alkydes saponifient sur l'acier galvanisé et sur les métaux non-ferreux.
16. Ne jamais appliquer un liant saponifiable sur un support riche en zinc.

b) solvant.

1. Le Zincano E résiste à des solvants très agressifs qui sont les hydrocarbures aromatiques et esthers.
2. Un diluant cellulaire dissout les éléments synthétiques.
3. Un solvant ne doit pas agresser ce qui se trouve en-dessous.
Cas particulier : les solvants extérieurs ne doivent pas altérer la couche de finition.

c) pigment.

1. Dans la mesure du possible, il faut conserver le même type de liant pour tout le système.
2. L'oxyde de fer micacé offre une barrière à l'infiltration de l'humidité.
3. L'aluminium lamellaire offre une barrière à l'infiltration d'humidité.
4. Les pigments inhibiteurs de type chromate et phosphate de zinc assurent une protection de courte durée.
5. L'oxyde de titane à plus de 30% en volume assure une protection passive contre la corrosion.
6. Règles sur les pigments anti-corrosion.
 - le minium de plomb est toxique et de teinte orange.
 - le chromate de zinc ne convient pas dans un environnement acide (pH<4), et est de teinte jaune ou verte.
 - le phosphate de zinc est un pigment très actif, est translucide blanc et peut donc être teinté.
 - le plomb métal assure une protection passive et est de teinte grise.

4.2. CONCEPTUALISATION

d) règles de compatibilité générales.

1. Si un produit ne peut s'appliquer sur un support courant, alors il faut trouver un deuxième produit qui peut s'appliquer sur ce support ou qui en modifie l'état, et qui soit compatible avec le premier.
2. Si un traitement n'adhère pas sur le support courant, alors il faut trouver un deuxième traitement qui adhère sur ce support, ou qui modifie son état, et sur lequel le premier traitement adhère.

4.2.4.2. Règles sur les systèmes.

a) résistances.

1. La résistance déterminée par le type de support peut être moindre ou inexistante quand l'environnement est intérieur.
2. Un produit qui ne résiste à aucune atmosphère ou ambiance n'est pas un produit de finition.
3. Si le produit ne résiste pas, alors il faut le couvrir d'un produit qui résiste et qui adhère sur ce qui se trouve en-dessous.
4. Si un produit satisfait toutes les exigences fixées par l'utilisateur et dépendant de l'environnement, alors ce produit est une finition acceptable.

b) aspect.

1. Si l'aspect désiré est "peint", alors il faut une opacité égale à 1.
2. Si l'aspect désiré est "vernis", alors il faut que le traitement antérieur soit différent de peinture avec opacité égale à 1; sinon, il faut décaper.
3. Si l'aspect désiré est "vernis", alors on ne peut pas conseiller un primer opaque.
4. Si l'aspect désiré est non-filmogène, alors le traitement antérieur ne peut pas être filmogène; sinon, il faut décaper.
5. Un produit maigre ne peut pas s'appliquer sur un produit gras.

4.2. CONCEPTUALISATION

c) généralités.

1. Un système anti-corrosion doit avoir une épaisseur totale minimale de 200 microns.
2. Un nombre plus grand de couches minces donne généralement un meilleur résultat qu'un nombre plus petit de couches épaisses.
3. Un nombre trop important de couches peut accélérer la détérioration de l'aspect final.

4.2.4.3. Règles portant sur le type de support.

1. Si la nature du support est de l'aluminium, alors il faut modifier l'état de surface du support, et faire un essai car il existe différents types d'aluminium.
2. Si la nature du support est de l'acier galvanisé, alors :
 - il n'est pas nécessaire d'avoir des pigments anti-corrosion; une couche épaisse peut suffire.
 - une peinture riche en zinc est une très bonne solution mais n'est pas indispensable.
 - on peut appliquer une peinture à l'eau ou une peinture dont le liant est le caoutchouc chloré, mais jamais un alkyde sans traitement préalable.
 - on peut appliquer une peinture dont le liant est un alkyde à condition d'appliquer d'abord un primer (par exemple: Washprimer).
3. Si la nature du support est de l'acier, alors:
 - les peintures dont les liants sont des alkydes ou des caoutchoucs chlorés sont conseillées.
 - les peintures à l'eau sont peu conseillées.
 - il faut absolument une protection anti-corrosion.

4.3. FORMALISATION

4.3. FORMALISATION.

Au cours de cette phase, nous n'avons pas suivi la méthode de HAYES-ROTH décrite au chapitre 3. En effet, elle nous a paru fort vague et peu adéquate pour le genre de problème dont nous avons à traiter en ce sens qu'elle n'incorpore pas la notion de chronologie dans la recherche d'une solution. Nous avons donc choisi une autre découpe des activités relatives à la tâche de formalisation.

Nous commencerons par présenter le modèle que nous avons retenu pour représenter le problème de la mise en peinture. Puis, nous formaliserons les concepts de support et de traitement ainsi que la relation qui les unit en termes de ce modèle.

4.3.1. MODELISATION DU PROBLEME DE MISE EN PEINTURE.

Nous avons pensé modéliser la connaissance sous forme d'un schéma entité/relation [BOD83]. Il est apparu assez rapidement que ce schéma ne présentait pas suffisamment de souplesse dans sa forme, et qu'en particulier, il était impossible d'y représenter une évolution temporelle des connaissances et d'y intégrer une association entité-action. Un schéma REMORA [ROL82] aurait sans doute été plus adapté pour remédier à ce problème, puisqu'il permet de séparer les informations relatives à un même objet, mais ayant des périodicités différentes.

Finalement, nous avons choisi une modélisation assez classique en Intelligence Artificielle, à savoir la modélisation d'un problème par espace d'états [NIL71].

Dans cette modélisation, il y a deux concepts importants : les états et les opérateurs. Le processus de recherche ne modifie pas les configurations physiques du problème mais il travaille sur une description de ces configurations.

Dans cette approche "Espace d'états", le problème peut être décrit par le quadruplet $\langle S, I, F, OP \rangle$ où

S est un ensemble d'états;

I est un ensemble d'états initiaux représentant les données du problème et leurs propriétés ($I \subset S$);

F est un ensemble d'états finaux représentant la solution cherchée et ses propriétés ($F \subset S$);

OP est un ensemble d'opérateurs de transformation d'états
($OP : S \rightarrow S$).

Dès lors, résoudre un problème revient à déterminer une suite d'états intermédiaires S_i tels que

$S_0 \in I, S_n \in F,$

$S_i = OP_k (S_{i-1})$ où OP_k est applicable à S_{i-1} .

La solution au problème est constituée de la suite d'opérateurs OP_1, \dots, OP_n tels que

$S_n = OP_n (OP_{n-1} (\dots (OP_1 (S_0)) \dots))$ où $S_0 \in I$ et $S_n \in F$.

4.3. FORMALISATION

La méthode de résolution consiste à générer de nouvelles descriptions, S_i , à partir d'anciennes, S_{i-1} , par application d'un opérateur OP_k et ensuite à vérifier si S_i a les propriétés décrites par S_n : c'est le GENERATE AND TEST.

Cette méthode est souvent utilisée dans les problèmes de planification et de robotique et nous allons l'illustrer sur un exemple pris dans ce dernier domaine [NIL71].

Le problème est le suivant : un singe est dans une pièce contenant une boîte et un régime de bananes. Les bananes sont suspendues et hors d'atteinte du singe; quelle séquence d'actions le singe devra-t-il exécuter pour obtenir les bananes ?

La représentation de ce problème doit contenir les éléments suivants : la position du singe dans la pièce (horizontalement et verticalement), la position de la boîte dans la pièce et une indication du fait que le singe a les bananes ou pas.

Ces éléments peuvent être représentés par le quadruplet (w,x,y,z) où w : position horizontale du singe,
 x : 1 ou 0 si le singe est sur la boîte ou pas, respectivement,
 y : position horizontale de la boîte,
 z : 1 ou 0 si le singe a les bananes ou pas, respectivement.

Les opérateurs correspondant aux quatre actions que le singe peut accomplir sont :

1. aller(u) : le singe va à la position u (u est une variable),
2. pousser(v) : le singe pousse la boîte en v (v est une variable),
3. grimper : le singe monte sur la boîte,
4. saisir : le singe saisit les bananes.

L'applicabilité et les effets des opérateurs sont donnés par les règles suivantes :

•
 $(w,0,y,z) \text{ ----aller}(u)\text{ ----> } (u,0,y,z)$
 $(w,0,w,z) \text{ ----pousser}(v)\text{ ----> } (v,0,v,z)$
 $(w,0,w,z) \text{ ----grimper}\text{ ----> } (w,1,w,z)$
 $(c,1,c,0) \text{ ----saisir}\text{ ----> } (c,1,c,1)$ où c est la position au sol juste en-dessous des bananes.

Pour notre part, nous avons choisi de représenter les éléments de l'espace d'états grâce au formalisme de vecteurs.

Les états, dans notre cas, les descriptions du support, sont représentés par des agrégats dont les différentes composantes sont les caractéristiques mises en évidence à la section 4.2.2.; plus précisément, l'état d'un support sera représenté par un VECTEUR D'ETAT, dont les différentes composantes représentent des valeurs pour les caractéristiques correspondantes du support.

4.3. FORMALISATION

renseigne pas de solutions. A ce moment, il faut négocier avec l'utilisateur pour supprimer une ou plusieurs exigences.

Dès lors, nous avons décidé de construire le système de peinture en partant des exigences décrites par l'utilisateur pour arriver au support décrit initialement, ce qui signifie que le système de peinture généré jusqu'alors est une solution au problème posé. Ce raisonnement est donc régressif.

Signalons dès à présent que l'implémentation de ce processus de raisonnement est facilitée par PROLOG. En effet, PROLOG adopte une stratégie de recherche en profondeur d'abord, en raison du fait que son moteur d'inférence travaille par chaînage arrière. De plus, cette recherche est exhaustive, puisque le mécanisme de "backtracking" permet à PROLOG de défaire les instanciations qu'il a déjà faites pour recherche de nouvelles solutions. Nous reviendrons sur ces éléments à la section 4.4., qui traite de l'implémentation.

4.3.3. ETUDE DETAILLEE DES COMPOSANTES DU MODELE.

4.3.3.1. Les supports.

Le modèle représentant le support à peindre est directement issu du concept de SUPPORT. Etant donné le choix de modélisation, ce concept est représenté par un VECTEUR D'ETAT, et ses caractéristiques constituent les attributs du vecteur d'état.

Le seul changement apporté se situe au niveau de la caractéristique "Traitements déjà subis" qui n'est plus retenue dans notre sous-système; l'attribut du vecteur d'état correspondant, "Dernier traitement vers le haut" provient du fait que le système est construit de manière régressive en partant de la finition pour aller vers le support dans son état initial.

Le vecteur d'état d'un support se compose de neuf composantes, dont certaines peuvent être répétitives, décomposables ou obligatoires.

Les éléments qui ne sont pas obligatoires recevront une valeur par défaut particulière par composante.

La signification des valeurs pour les différentes composantes est celle utilisée couramment dans le jargon du monde de la peinture. Il est cependant utile de préciser le sens de certaines d'entre elles.

En ce qui concerne l'attribut "nature", le sens de sa valeur ne pose pas de problème.

Les valeurs des attributs "forme" et "résistance" (à l'exception de celles concernant l'environnement sur lesquelles nous reviendrons ci-après) sont moins strictement définies et davantage laissées à l'appréciation de l'utilisateur.

Les valeurs de l'attribut "environnement" sont définies par la norme AFNOR qui se trouve en annexe C.

4.3. FORMALISATION

Les valeurs des attributs "caractéristiques spéciales" et "protections" sont celles utilisées dans les fiches de description technique des produits T.P.C.

Les valeurs de l'attribut "état de surface" sont laissées à l'appréciation de l'utilisateur sauf en ce qui concerne le niveau de corrosion qui est défini par la norme suédoise reprise en annexe C.

Les valeurs de l'attribut "aspect", sans être définies formellement, représentent des standards dans le monde de la peinture.

Les valeurs possibles pour chaque composante du vecteur d'état ont été classées selon deux manières non-exclusives.

1. Quand il y a une gradation dans les exigences représentées par chaque valeur, elles sont données dans l'ordre décroissant.

2. Un système de classes a été établi pour répertorier les différentes valeurs.

Classe a) : valeurs à supprimer, c'est-à-dire qui n'apparaissent plus dans E_i si elles sont spécifiées dans la postcondition du traitement TRT_i ; sinon, elles sont maintenues dans E_i .

Classe b) : valeurs à remplacer, c'est-à-dire qui doivent apparaître nécessairement dans la postcondition du traitement TRT_i si elles sont spécifiées dans E_i .

Classe c) : valeurs permanentes, c'est-à-dire qui doivent être spécifiées dans chacun des E_i ($1 \leq i \leq n$).

Les neuf composantes sont les suivantes :

1. **Nature du support** : obligatoire et non-répétitif.

Valeurs possibles :

- acier	----+
- acier galvanisé	----!
- aluminium	----+----- métaux
- cuivre	----!
- zinc	----+

2. **Forme du support** : facultatif et répétitif.

Valeur par défaut : correspond au niveau d'exigence le plus bas, à savoir plan, horizontal et surface en haut.

N.B. Les valeurs sont classées par ordre décroissant d'exigence dans chaque sous-classe.

4.3. FORMALISATION

Valeurs possibles :

- vertical ou oblique ou horizontal
- tubulaire ou angulaire ou plan
- surface vers le bas ou surface vers le haut

3. Environnement du support : obligatoire et non-répétitif.

N.B. Les valeurs sont classées par ordre décroissant d'exigence dans chaque sous-classe.

Valeurs possibles :

- extérieur: - atmosphère marine
- atmosphère industrielle
- atmosphère urbaine
- atmosphère rurale

- intérieur: - ambiance très agressive
expl : hall de fabrication
- ambiance moyennement agressive
expl : cuisine, salle de bains
- ambiance peu agressive
expl : salon

- température de l'air ambiant : de -20°C à 40°C
- taux d'humidité de l'air ambiant : de 0% à 100%

4. Caractéristiques spéciales : facultatif et répétitif.

Valeur par défaut : indifférent; signifie qu'aucune caractéristique spéciale n'est exigée.

Valeurs possibles :

- classe b) : - non-toxique : OUI/NON
- antidérapant : OUI/NON
 - peu salissant : OUI/NON

- classe c) : - sans coulures : OUI/NON

4.3. FORMALISATION

5. Etat de surface : obligatoire, décomposable et répétitif.

Valeurs possibles :

classe b) : - dégradé : - corrodé : - peu rouillé
- voile de rouille adhérente
- rouille importante
- oxydé

- gras
- poussiéreux
- sale
- mauvais état (écaillé ou farinant)
- n'accroche pas

- recouvrable : - non-corrodé
- non-gras
- non-poussiéreux
- propre
- net : - en bon état
- mis à nu
- neuf
- accrochant

6. Aspect : facultatif et répétitif.

Valeur par défaut : indifférent; signifie que l'aspect peut être quelconque sans restriction.

Valeurs possibles :

classe a) : - opaque
- coloré
- structuré
- filmogène

classe b) : - brillant
- satiné
- mat
- spécial : - pailleté
- métallisé
- craquelé
- martelé

classe c) : - translucide
- incolore
- lisse
- non-filmogène

4.3. FORMALISATION

7. Résistance : facultatif et répétitif.

Valeur par défaut : aucune; signifie qu'aucun type de résistance n'est assuré.

Valeurs possibles :

N.B. Les valeurs sont classées par ordre décroissant d'exigence dans chaque sous-classe.

classe b) : - à l'environnement (UV, agressions atmosphériques, ruissellement de l'eau)

a1. atmosphère marine

b1. " industrielle

b2. " urbaine

b3. " rurale

c1. ambiance très agressive

c2. " moyennement agressive

c3. " peu agressive

- à un milieu aqueux (stagnation d'eau)

- au lavage (abrasion moyenne + détergents ménagers + eau)

- à l'abrasion mécanique :

e1. abrasion légère (usure dans une pièce habitée)

e2. " moyenne (griffes, frottement, trafic lég)

e3. " forte (trafic lourd)

- aux traces de doigts

- au contact de la terre

- au feu (=ignifugation)

classe c) : - à l'infiltration de vapeur d'eau

- aux variations de température (bornes inf. et sup.)

8. Protections : facultatif et répétitif.

Valeur par défaut : aucune; signifie qu'aucun type de protection n'est assuré.

Valeur possible :

classe a) : contre la corrosion

4.3. FORMALISATION

9. Nom du dernier traitement vers le haut : facultatif pour E1 et En, et obligatoire pour les Ei ($1 < i < n$) et non-répétitif.

Valeur par défaut : aucun; signifie qu'aucun traitement vers le haut n'est défini.

Valeur possible :

- un des traitements existants (voir liste)

4.3.3.2. Les traitements.

Un TRAITEMENT est une formalisation du concept de même nom et de la relation MODIFIE.

Les caractéristiques du concept provenant de la phase de conceptualisation ont été remaniées lors de la formalisation. La structure du traitement était relativement complexe puisqu'elle portait, parfois pour un même attribut, sur deux occurrences du concept support. Par exemple, la caractéristique "Conditions d'application" d'un traitement concernait l'attribut "Caractéristiques spéciales" de l'occurrence de support représentant un support avant application du traitement alors que sa propre composante "Caractéristiques spéciales" portait sur le même attribut du concept support mais était relatif à l'occurrence représentant le support après traitement.

Pour simplifier les choses, deux occurrences du concept de support séparées chronologiquement mais apparaissant dans une même occurrence de traitement seront représentées de façon distincte. C'est pourquoi, de manière explicite, les caractéristiques du concept de traitement ont été modélisées sous la forme de deux vecteurs d'état du support, l'un représentant l'occurrence du support avant application du traitement (PRECONDITION), l'autre représentant celle après application du traitement (POSTCONDITION).

De plus, deux types de renseignement accompagnent la représentation des opérateurs, à savoir la composition chimique et les caractéristiques d'application du traitement. En effet, les caractéristiques d'application du traitement n'ont pas de raison de se trouver dans les préconditions ou postconditions d'un traitement, étant donné qu'elles ne constituent pas des critères de décision dans la construction d'un système de peinture. Notons cependant que des versions ultérieures du système CONSEIL pourraient intégrer des critères faisant partie de cette caractéristique. Par exemple, le temps total d'application du système, c'est-à-dire la somme des temps de séchage et d'application de tous les traitements, peut être un facteur de choix parmi les différentes solutions. En ce qui concerne la composition chimique des traitements, on peut en déduire certaines caractéristiques du vecteur d'état. Cette possibilité n'a pas été implémentée ici, mais l'option que nous avons prise permettra d'éviter un certain

4.3. FORMALISATION

nombre de redondances, en donnant la possibilité au système d'intégrer dynamiquement dans les vecteurs d'état les caractéristiques déductibles de la composition chimique.

La description de l'opérateur traitement se compose donc de cinq éléments qui sont les suivants :

1. Nom : obligatoire et non-répétitif.
2. Eléments obligatoires du vecteur d'état avant traitement :
obligatoire et non-répétitif

= PRECONDITION DE L'OPERATEUR
3. Eléments du vecteur d'état modifiés par le traitement :
obligatoire et non-répétitif

= POSTCONDITION DE L'OPERATEUR
4. Composition : obligatoire pour les traitements de la gamme T.P.C. et facultatif pour les traitements de type préparation; non-répétitif.

5. Descriptif du traitement :

- Matériel et dilution : - matériel : obligatoire
- dilution : facultatif

Le groupe est répétitif.

Valeur par défaut pour dilution : non-dilué; signifie qu'il ne faut pas diluer avec ce matériel.

- Intervalles de recouvrement : facultatif et non-répétitif.
 - minimal
 - maximal

Valeur par défaut : quelconque; signifie qu'il n'y a pas d'intervalles de temps maximum ou minimum à respecter pour recouvrir le traitement

- Température et taux d'humidité de l'air ambiant :
facultatif et non-répétitif

Valeur par défaut : - température : quelconque
- taux d'humidité : quelconque

- Epaisseur du traitement : facultatif et non-répétitif.
Valeur par défaut : 0.

- Nombre d'applications du traitement : facultatif et non-répétitif.

Valeur par défaut : 1.

4.3. FORMALISATION

4.3.4. CONSTRUCTION REGRESSIVE DE LA SUITE DE TRAITEMENTS.

Comme esquissé à la section 4.3.1., la suite de traitements conseillée est obtenue par un processus de recherche dans l'espace des vecteurs d'états du problème. Il s'agit de construire la suite $E_0, E_1, \dots, E_i, \dots, E_n$ où $E_0 = E_I$ est le vecteur d'état initial et $E_n = E_F$ est le vecteur d'état final et $E_i = TMT_i (E_{i-1})$ où TMT_i est le i ème traitement à appliquer dans la suite solution ($1 \leq i \leq n$).

Pour les raisons avancées en 4.3.2., cette suite est construite de manière **régressive** en partant de l'état final E_n . Deux problèmes sont maintenant abordés successivement : comment sélectionner un traitement TMT_i pour l'ajouter à la suite solution obtenue jusqu'alors et à partir de là, comment déterminer le nouveau vecteur d'état intermédiaire E_{i-1} à partir de E_i et TMT_i .

4.3.4.1. Sélection d'un traitement.

4.3.4.1.1. Principe.

Comme il l'a été expliqué au point 4.3.2., le principe est de comparer l'état courant aux états spécifiés dans la postcondition des traitements candidats, c'est-à-dire résultant de l'application de ces derniers.

D'une part, des règles générales de sélection d'un traitement (c'est-à-dire ne portant pas sur un traitement particulier) ont été formalisées. Elles se décomposent en deux familles selon les attributs du vecteur d'état qu'elles manipulent.

Pour ce qui est de la première famille, les règles de sélection sont immédiates et relativement simples. En effet, elles sont le pendant direct des correspondances établies pour la relation "modifie" au niveau de la conceptualisation. Elles portent sur les attributs "nature", "environnement", "forme" et "caractéristiques spéciales" du vecteur d'état; il est nécessaire que la postcondition du traitement spécifie pour ces attributs des valeurs identiques ou plus contraignantes que celles spécifiées par le vecteur d'état courant.

Les règles de la seconde famille ont pour objectif principal de vérifier que le traitement sélectionné est pertinent dans le système de peinture en construction, c'est-à-dire qu'il apporte bien un élément de réponse au problème de l'utilisateur. On évite ainsi de sélectionner des traitements qui ne nuisent pas au système de peinture mais n'en sont pas pour autant indispensables. Elles concernent les attributs suivants du vecteur d'état : "aspect", "résistances" et "protections".

D'après ces règles, un traitement convient si les valeurs spécifiées par les attributs "aspect" et "résistances" du vecteur d'état du support courant, qui appartiennent aux classes permanente et à remplacer, sont également spécifiées par les attributs correspondant dans la postcondition du traitement.

4.3. FORMALISATION

D'autre part, des règles relatives à des traitements particuliers ont été élaborées et ce, toujours dans le but de ne sélectionner que les traitements indispensables dans la suite solution. Elles font référence à l'état initial du support, anticipant en quelque sorte le mécanisme de construction. A titre d'illustration, citons le cas du traitement "mordantage" qui n'apparaîtra dans la suite solution que si la valeur "n'accroche pas" pour la composante "état de surface" est spécifiée dans le vecteur d'état initial.

Finalement, la dernière règle de sélection d'un traitement porte sur la compatibilité chimique entre le traitement candidat et le dernier traitement vers le haut du support courant. La compatibilité chimique est immédiate entre un traitement de type préparation et un autre traitement. Dans les autres cas, elle est déterminée par les conditions suivantes, avec les conventions d'écriture rappelées ci-dessous :

EF = En : état final;
EI = Eo : état initial;
 $TRTi (E_{i-1}) = E_i$ et $1 \leq i \leq n$.

- 1) concernant le PVC (pigment volume concentration) : la valeur du PVC doit décroître du support vers l'état final :

$$PVC(\text{finition}) \leq PVC(\text{interméd}) \leq PVC(\text{primer})$$

- 2) concernant l'agressivité des solvants (A) : l'agressivité du solvant doit décroître du support vers l'état final :

$$A(\text{finition}) \leq A(\text{interméd}) \leq A(\text{primer})$$

- 3) concernant le caractère thermoplastique des traitements : la thermoplasticité du liant doit croître du support vers l'état final :

soit TG , la température de transition vitreuse,

si TRT_j = produit thermoplaste,
alors TRT_i = produit thermoplaste pour $j < i$
et
 $TG(TRT_k) < TG(TRT_j)$ pour $1 \leq j < k \leq i$

et si TRT_i = produit duroplaste
alors TRT_{i+1} = produit thermoplaste ou duroplaste

4.3. FORMALISATION

4) concernant les liants

Règle générale : dans la mesure du possible, le liant doit rester constant dans le système de peinture :

$$\text{liant}(\text{TRTi}) = \text{liant}(\text{TRTi}+1), 1 \leq i < n$$

On peut aussi avoir :

- a) liant(TRTi) = caoutchouc chloré
et liant(TRTj) = copolymères vinyliques chlorés
avec $i = j+1$ ou $i = j-1$
- b) si liant(TRTi) = résines alkydes
alors liant(TRTi+1) = résines alkydes
ou caoutchouc chloré
ou huiles siccatives
ou résines oléostyrénées
- c) si liant(TRTi) = résines époxydiques
alors liant(TRTi+1) = résines époxydiques
ou " acryliques
ou " alkydes
ou caoutchouc chloré
ou copolymères vinyliques chlorés
ou PUR mono aliphatiques
ou " " aromatiques
ou " bicomposants
ou émulsions aqueuses
- d) si liant(TRTi) = huiles siccatives
ou polymères azotés
ou polyvinyl butyral
ou résines alkydes
ou " oléostyrénées
ou " phénoliques
alors ((i est différent de 1)
ou (nature(support) n'est pas galva
et n'est pas zinc))
ou
(pigment(TRTi-1) n'est pas borate de zinc et
n'est pas chromate de zinc et
n'est pas phosphate de zinc et
n'est pas oxyde de zinc et
n'est pas zinc métal)

5) concernant le décapage

Quand une incompatibilité entre produits force le décapage, il faut redéfinir un nouvel état initial avec la valeur "nu" dans la composante "état de surface".

4.3. FORMALISATION

4.3.4.1.2. Synthèse des règles de sélection.

On dispose du vecteur d'état courant, E_i et d'un traitement candidat, TRT_i , avec sa postcondition, $POST_i$.

Conventions d'écriture :

On représente par le nom de la composante écrit en majuscules la valeur de cette composante du vecteur d'état.

ex. NATURE [E_i] représente la valeur de la composante "nature" du vecteur d'état E_i ,

FORME [$POST_i$] représente la valeur de la composante "forme" dans la postcondition du traitement TRT_i .

Le traitement candidat sera sélectionné et retenu aux conditions suivantes :

- 1) NATURE [E_i] est inclus dans NATURE [$POST_i$],
- 2) FORME [E_i] représente une exigence moins forte que celle représentée par FORME [$POST_i$],
- 3) ENVIRONNEMENT [E_i] représente une exigence moins forte que celle représentée par ENVIRONNEMENT [$POST_i$],
- 4) CARACTSPEC [E_i] est inclus dans CARACTSPEC [$POST_i$],
- 5) Les valeurs appartenant aux classes permanente et à remplacer de ASPECT [E_i] sont incluses dans ASPECT [$POST_i$],
- 6) ETATSURF [$POST_i$] est inclus dans ETATSURF [E_i],
- 7) RESIST [E_i] représente une exigence moins forte que celle représentée par RESIST [$POST_i$],
- 8) il n'y a pas de condition portant sur PROTECTION,
- 9) DERNTRT [E_i] est compatible au niveau chimique avec TRT_i .

4.3.4.2. Passage au nouveau vecteur d'état.

4.3.4.2.1. Principe.

Les trois premières composantes du vecteur d'état sont invariables. Dès lors, elles sont fixées par la description des états final et initial et ne seront plus jamais modifiées par la suite.

La dernière composante du nouvel état courant qui est "dernier traitement vers le haut", aura pour valeur le nom du traitement qui a été retenu.

Pour toutes les autres composantes, la modification se fait en fonction des classes de valeurs mentionnées à la section 4.3.3.1. et de la précondition du traitement retenu qui se retrouve d'office dans la nouvelle description du support courant. Par définition des classes de valeur, les modifications se font de la façon suivante pour chaque classe :

si une composante a une valeur appartenant à la classe a), alors cette valeur est supprimée si elle se trouve dans la postcondition du

4.3. FORMALISATION

traitement appliqué:

si une composante a une valeur appartenant à la classe b), alors cette valeur est remplacée dans tous les cas par la valeur de la composante correspondante dans la précondition;

si une composante a une valeur appartenant à la classe c), alors cette valeur est invariante, c'est-à-dire qu'elle ne peut disparaître du vecteur d'état et doit être respectée par tous les traitements. Si un traitement a dans sa précondition une autre valeur pour la même composante, alors cette valeur est ajoutée à la première.

4.3.4.2.2. Synthèse des règles de modification.

On dispose du vecteur d'état courant, E_i , du traitement retenu, TRT_i , avec sa postcondition, $POST_i$ et sa précondition, PRE_i à partir desquels il faut calculer le nouveau vecteur d'état E_{i-1} .

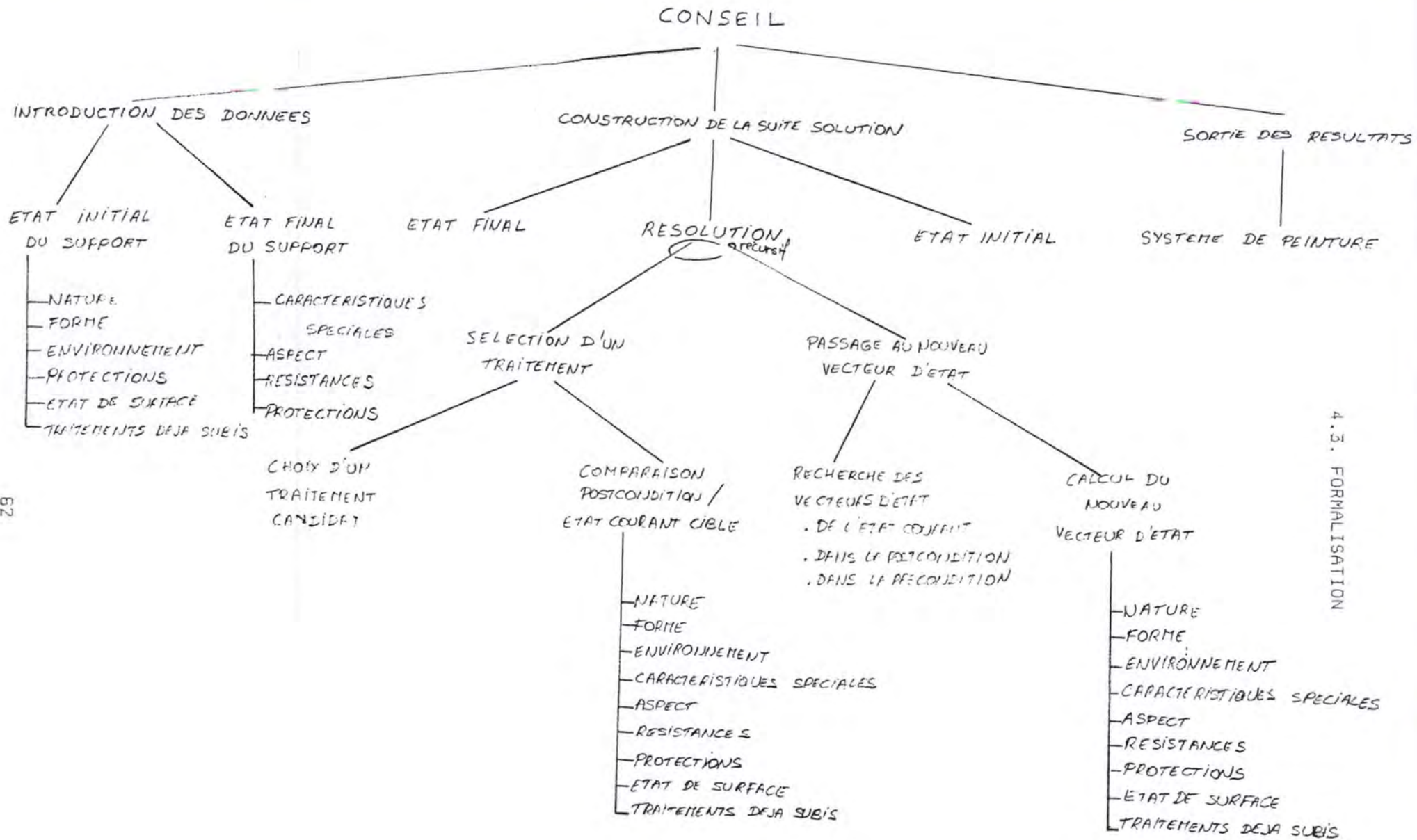
Conventions d'écriture :

Elles sont les mêmes que celles décrites au point 4.3.4.1.2.

Les règles de calcul des valeurs des composantes du nouvel état courant sont les suivantes :

- 1) NATURE $[E_{i-1}] = \text{NATURE } [E_i]$,
- 2) FORME $[E_{i-1}] = \text{FORME } [E_i]$,
- 3) ENVIRONNEMENT $[E_{i-1}] = \text{ENVIRONNEMENT } [E_i]$,
- 4) CARACTSPEC $[E_{i-1}] = \text{CARACTSPEC } [PRE_i]$,
- 5) ASPECT $[E_{i-1}] = \text{PERM (ASPECT } [E_i]) +$
ASPECT $[PRE_i] +$
(SUPP (ASPECT $[E_i]$) -
SUPP (ASPECT $[POST_i]$)),
- 6) ETATSURF $[E_{i-1}] = \text{ETATSURF } [PRE_i]$,
- 7) RESIST $[E_{i-1}] = \text{PERM (RESIST } [E_i]) +$
RESIST $[PRE_i] +$
(RESIST $[E_i]$ -
RESIST $[POST_i]$),
- 8) PROTECTION $[E_{i-1}] = \text{PROTECTION } [PRE_i] +$
(PROTECTION $[E_i]$ -
PROTECTION $[POST_i]$),
- 9) DERNTRT $[E_{i-1}] = TRT_i$.

La page suivante reprend le nouvel affinage ainsi développé du diagramme des tâches obtenu au point 4.2.3.



4.3. FORMALISATION

4.4. IMPLEMENTATION

4.4. IMPLEMENTATION.

Comme il l'a été exposé au point 4.2.1., nous avons pris la décision de nous limiter dans un premier temps au sous-système constitué par les supports de type métallique écrit en PROLOG.

Nous avons déjà dit dans la phase d'identification que nous n'avons pas eu à choisir le langage d'implémentation puisqu'il était prévu dès le départ que nous écririons le prototype en PROLOG [CL084], [GIA85], [KLU85].

C'est un langage qui a été conçu par le groupe de recherche en Intelligence Artificielle de Marseille, avec à sa tête Alain Colmerauer, dans le cadre de recherches sur la compréhension du langage naturel.

Nous allons brièvement et, de façon simplifiée, rappeler le mode de fonctionnement de PROLOG. Un système PROLOG est basé sur un démonstrateur de théorème par le Principe de Résolution de Robinson [ROB68] pour clauses de Horn. La stratégie particulière utilisée est une forme de résolution linéaire : on commence par résoudre le but avec l'une des hypothèses pour donner une nouvelle clause. Puis on résout la dernière clause obtenue avec l'une des hypothèses originales. En termes de PROLOG, on peut considérer la dernière clause dérivée comme la conjonction des buts qui restent à satisfaire. Ceci démarre par la question et se termine, si tout va bien par la clause vide. À chaque étape nous trouvons une clause dont la tête coïncide avec l'un des buts, on instancie les variables nécessaires, on élimine le but que l'on a unifié, et on ajoute le corps de la clause instanciée aux buts à satisfaire.

En PROLOG, on s'occupe toujours du premier des littéraux de la clause but. De plus, les nouveaux buts dérivés de l'usage d'une clause sont placés en tête de la clause but, ce qui signifie que PROLOG finit de satisfaire un sous-but avant d'aller essayer quoi que ce soit d'autre. La stratégie de recherche parmi les autres clauses, pour satisfaire ce but, est une recherche en profondeur d'abord. Ceci veut dire qu'il ne considère qu'une seule alternative à la fois, suivant les implications en supposant que le choix est correct. Pour chaque but, il choisit les clauses dans un ordre fixé et il n'en vient à considérer les dernières que si toutes les précédentes n'ont pas réussi à produire de solution.

PROLOG offre donc un formalisme assez transparent pour représenter des connaissances de type "calcul des prédicats de premier ordre" et présente l'avantage de posséder un algorithme de recherche préprogrammé dans un arbre ET/OU en profondeur d'abord.

4.4.1. STRATEGIE GLOBALE.

CONSEIL utilise donc les clauses de Horn comme mode de représentation des connaissances et le principe de résolution pour les manipuler. Le concept d'unification est particulièrement bien adapté aux exigences de notre système puisque

4.4. IMPLEMENTATION

dans la construction d'un système de peinture, il s'agit d'évaluer la pertinence du choix de certains traitements et de considérer les conditions de leur sélection. Ce processus est assez facilement décomposable en sous-tâches et peut être représenté sous la forme d'un arbre ET/OU.

L'approche adoptée dans CONSEIL, de type descendante et régressive, est assez semblable à celle utilisée dans MYCIN; on démarre avec un but (construire un système de peinture) et on génère alors les sous-buts dont la solution doit résoudre le problème de départ.

Conceptuellement, il y a deux types de connaissances dans CONSEIL :

- la connaissance relative aux traitements et
- la connaissance relative aux mécanismes de raisonnement pour sélectionner un ensemble de traitements constituant un système de peinture.

Ces deux types de connaissances sont représentés par un ensemble de règles PROLOG qui diffèrent par le fait que celles représentant la connaissance relative aux traitements ont pour paramètres des constantes alors que dans la deuxième catégorie, apparaissent surtout des variables.

Il peut arriver que CONSEIL puisse, à un moment donné, sélectionner plus d'un traitement (on peut considérer qu'il y aurait un conflit entre traitements). Il n'est pas prévu, dans l'état actuel des choses, de stratégie de résolution de conflit étant donné que, lors de l'analyse, il a été demandé de générer tous les systèmes possibles pour le problème posé. Cette génération exhaustive est rendue possible en exploitant le backtracking sous-jacent à PROLOG. Dans l'avenir, on peut penser à un filtre permettant d'établir une échelle de qualité entre les solutions générées.

4.4.2. DEFINITION DES OBJETS MANIPULES.

4.4.2.1. Les types d'objet.

4.4.2.1.1. Structure.

Les types d'objet manipulés dans le programme PROLOG proviennent directement de la modélisation par espace d'états (cfr section 4.3); ils ont été réorganisés de manière légèrement différente pour faciliter l'implémentation PROLOG. Leurs définitions et caractéristiques sont les suivantes :

4.4. IMPLEMENTATION

TRAITEMENT : une préparation de support ou une application d'un produit de la gamme T.P.C.

COMPOSITION : description de la composition chimique d'un TRAITEMENT en termes de 4 COMPOSANTS.

COMPOSANT : solvant ou liant ou pigments entrant dans la composition d'un TRAITEMENT, ou encore taux de concentration de pigments en volume (Pvc).

NOM DU TRAITEMENT : identifiant d'un TRAITEMENT, c'est-à-dire son nom ou son appellation commerciale.

DESCRIPTIF : renseignements divers relatifs à un TRAITEMENT (mode d'application, conditions de conservation, etc).

PRECONDITION : VECTEUR D'ETAT représentant un support avant l'application du TRAITEMENT.

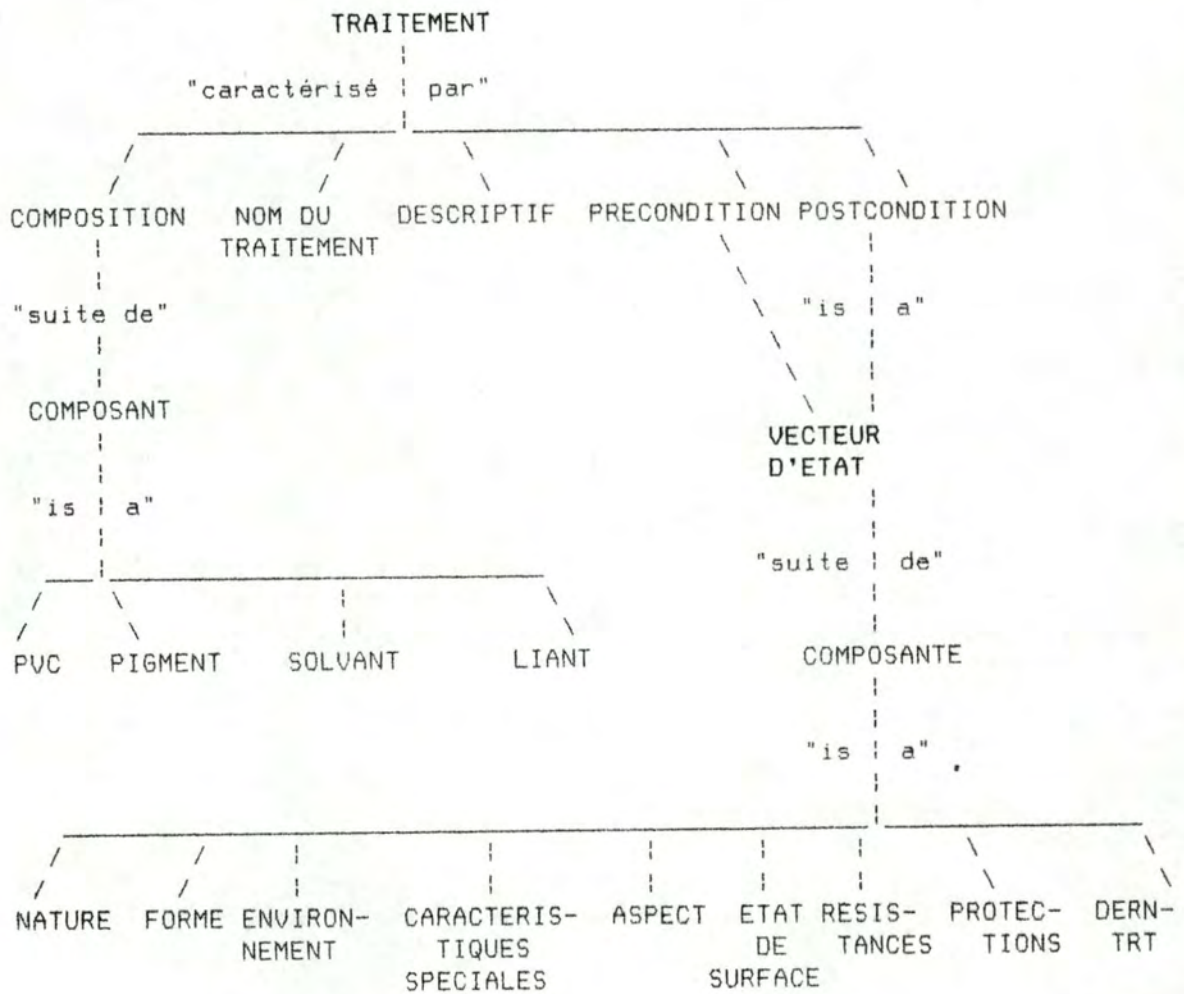
POSTCONDITION : VECTEUR D'ETAT représentant un support après l'application du TRAITEMENT.

VECTEUR D'ETAT : description d'un support à peindre, dans un état donné, en termes de 9 COMPOSANTES.

COMPOSANTE : nature ou forme ou environnement ou caractéristiques spéciales ou aspect ou état de surface ou résistances ou protections ou dernier traitement vers le haut, d'un support.

Ces différents types d'objet sont dès lors structurés selon le schéma de la page suivante :

4.4. IMPLEMENTATION



4.4.2.1.2. Primitives de manipulation associées à ces types d'objets.

Les spécifications de ces primitives sont données aux points 4.4.4. et 4.4.5.

Types d'objet	Procédure PROLOG
Traitement	POST PRE COMP EXTPVC EXTPIG EXTLIANT EXTSOL
Solvant	EHELAGR
Liant	EHELTERM SAPONIFIABLE
Pigment	COMPZINC

4.4. IMPLEMENTATION

Vecteur d'état	EXTNAT EXTFORME EXTENU EXTCARSPEC EXTETATSURF EXTAS EXTRES EXTPROT EXTDERNTRT EXTITEMSVECT
Composante	CROISS ITEMVALCLASSE

4.4.2.2. Les types d'objet génériques.

4.4.2.2.1. Structure.

Les types d'objet sont des abstractions dont le rôle est d'accroître la lisibilité et la modifiabilité de l'implémentation de CONSEIL; ils sont définis de la façon suivante :

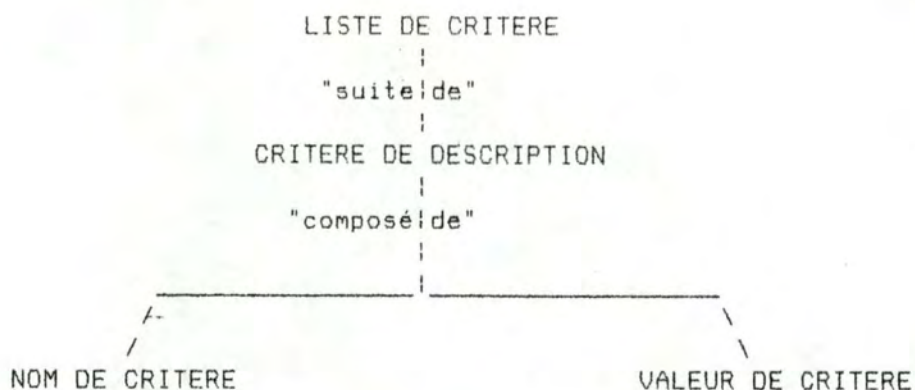
LISTE DE CRITERE : ensemble de CRITERES DE DESCRIPTION.

CRITERE DE DESCRIPTION : couple composé d'un nom de composante et de sa valeur.

NOM DE CRITERE : premier des deux éléments du couple CRITERE DE DESCRIPTION.

VALEUR DE CRITERE : second des deux éléments du couple CRITERE DE DESCRIPTION.

Les types génériques sont structurés selon le schéma suivant :



IMPLEMENTATION

4.4.2.2.2. Primitives de manipulation.

Les spécifications de ces primitives seront énoncées au point 4.4.5.

Types d'objet génériques	Règle PROLOG
Liste de critère	VALEUR_ITEM EXTVALCRIT
Valeur de critère	EXTLISTITEM

4.4.3. STRUCTURE DES REPRESENTATIONS INTERNES.

4.4.3.1. Définition de nouvelles structures PROLOG.

A côté des structures existantes de liste et d'atome, nous avons, pour des raisons de modifiabilité, défini de nouvelles structures utilisées pour l'implémentation des types d'objets et des types génériques.

CRITERE_DE_DESCRIPTION : liste de deux termes.

LISTE_DE_CRITERE : liste de CRITERE_DE_DESCRIPTIONs.

LISTE_DE_VALEUR : liste de VALEURs.

NOM_DE_CRITERE : atome.

NOM_DE_TRAITEMENT : atome.

VALEUR : atome.

LISTE_VIDE : représentation d'une liste ne contenant aucun terme.

4.4.3.2. Structures de représentation des types d'objet.

- Pour chaque occurrence d'un TRAITEMENT, la relation "caractérisé par" est représentée par quatre faits PROLOG. Dans chacune de celles-ci, le premier paramètre est un NOM_DE_TRAITEMENT qui représente NOM DE TRAITEMENT et identifie l'occurrence de TRAITEMENT.

Voici ces quatre faits sur lesquels nous reviendrons plus longuement en 4.4.4. ci-dessous :

```
COMP (NT, COMPOSITION),  
PRE (NT, PRECONDITION),  
POST (NT, POSTCONDITION),  
DESCRIP (NT, DESCRIPTIF).
```

IMPLEMENTATION

- COMPOSITION et VECTEUR D'ETAT sont représentés par des LISTES DE CRITERE et, par conséquent, COMPOSANT et COMPOSANTE sont implémentés sous forme de CRITERES DE DESCRIPTION.

4.4.3.3. Structures de représentation des types génériques.

- Les LISTES DE CRITERE sont implémentées sous forme de LISTES_DE_CRITERE et les CRITERES DE DESCRIPTION sous forme de CRITERE_DE_DESCRIPTION.

- Un NOM DE CRITERE est représenté par un NOM_DE_CRITERE.

- Une VALEUR DE CRITERE est représentée soit par une VALEUR si elle contient une seule valeur soit par une LISTE_DE_VALEUR si elle en contient plusieurs.

4.4.4. ARCHITECTURE DE LA BASE DE FAITS.

Nous faisons ici référence aux informations statiques concernant les traitements. Elles sont indispensables lors des phases de sélection d'un traitement candidat, de vérification de compatibilité avec la suite déjà construite et de calcul du nouvel état; elles portent sur les composants, la précondition et la postcondition d'un traitement.

Il est également nécessaire de disposer de certains renseignements sur ces informations elles-mêmes, c'est-à-dire sur les valeurs valides des COMPOSANTES et des COMPOSANTS et sur leurs caractéristiques.

Les FAITS représentant ces connaissances sont organisées en trois parties : la première est issue directement de la représentation interne du type d'objet TRAITEMENT, la deuxième traite de la hiérarchie des valeurs possibles des COMPOSANTES établie pour les valeurs des attributs du VECTEUR D'ETAT pendant la phase de formalisation et enfin, la troisième a pour objet les renseignements sur les COMPOSANTS.

a) Première classe de faits.

Les TRAITEMENTS : Les quatre règles citées au point 4.4.3.2. ont la sémantique définie ci-dessous :

- POST(Traitement, Vecteur).

où Traitement est un NOM_DE_TRAITEMENT,
Vecteur est une LISTE_DE_CRITERE.

Sémantique : Vecteur est la représentation interne de la POSTCONDITION du TRAITEMENT dont la représentation interne est Traitement.

4.4. IMPLEMENTATION

- PRE(Traitement, Vecteur).

où Traitement est un NOM_DE_TRAITEMENT,
Vecteur est une LISTE_DE_CRITERE.

Sémantique : Vecteur est la représentation interne de la PRECONDITION du TRAITEMENT dont la représentation interne est Traitement.

- COMP(Traitement, Composition).

où Traitement est un NOM_DE_TRAITEMENT,
Composition est une LISTE_DE_CRITERE.

Sémantique : Composition est la représentation interne de la COMPOSITION du TRAITEMENT dont la représentation interne est Traitement.

b) Deuxième classe de faits.

Hierarchie des valeurs possibles des COMPOSANTES: Rappelons que deux types de gradation avaient été établis lors de la formalisation pour classer les valeurs des attributs du vecteur d'état.

- L'ordre suivant le degré d'exigence représenté par les valeurs est implémenté par le fait spécifié ci-après.

CROISS(Item, Valeur).

où Item est un NOM_DE_CRITERE,
Valeur est une liste.

Sémantique : Valeur est la représentation interne de S, la suite ordonnée des VALEURS DE CRITERE possibles pour le CRITERE DE DESCRIPTION dont le NOM DE CRITERE est représenté par Item.

Soit $S = (S_1, S_2, \dots, S_i, \dots, S_n)$; pour i, j tels que $1 \leq i < j \leq n$, S_i est une VALEUR DE CRITERE moins contraignante que S_j .

- L'appartenance de valeurs à une classe est représentée par le fait spécifié ci-après.

ITEMVALCLASSE(Item, Valeur, Classe).

où Item est un NOM_DE_CRITERE,
Valeur est une liste,
Classe est un atome.

Sémantique : Valeur est la représentation interne des VALEURS DE CRITERE appartenant à la classe spécifiée par Classe et qui sont valides pour le CRITERE DE DESCRIPTION dont le NOM DE CRITERE est représenté par Item.

4.4. IMPLEMENTATION

c) Troisième classe de faits.

Les COMPOSANTS. Lors de la formalisation, nous avons expliqué comment la composition chimique d'un produit peut empêcher la sélection d'un traitement pour raison d'incompatibilité entre deux traitements successifs. Il fallait donc prévoir, dans la base de connaissances, des informations permettant d'évaluer cette compatibilité. Nous disposons de quatre types d'information sur les COMPOSANTS qui se traduisent dans les faits suivants :

- Echelle d'agressivité des solvants.

ECHELAGR(L).

où L est une LISTE_DE_CRITERE.

Sémantique : Chacun des NOM_DE_CRITERE de L est la représentation interne d'un solvant et la VALEUR_DE_CRITERE correspondant est la représentation interne du rang de ce solvant dans l'échelle d'agressivité des solvants.

- Echelle de thermoplasticité des liants.

ECHELTHERM(L).

où L est une LISTE_DE_CRITERE.

Sémantique : Chacun des NOM_DE_CRITERE de L est la représentation interne d'un liant et la VALEUR_DE_CRITERE correspondant est la représentation interne du rang de ce liant dans l'échelle de thermoplasticité des liants.

- Liants saponifiables.

SAPONIFIABLE(L).

où L est une liste.

Sémantique : Chaque élément de L est la représentation interne d'un liant saponifiable.

- Pigments composés de zinc.

COMPZINC(L).

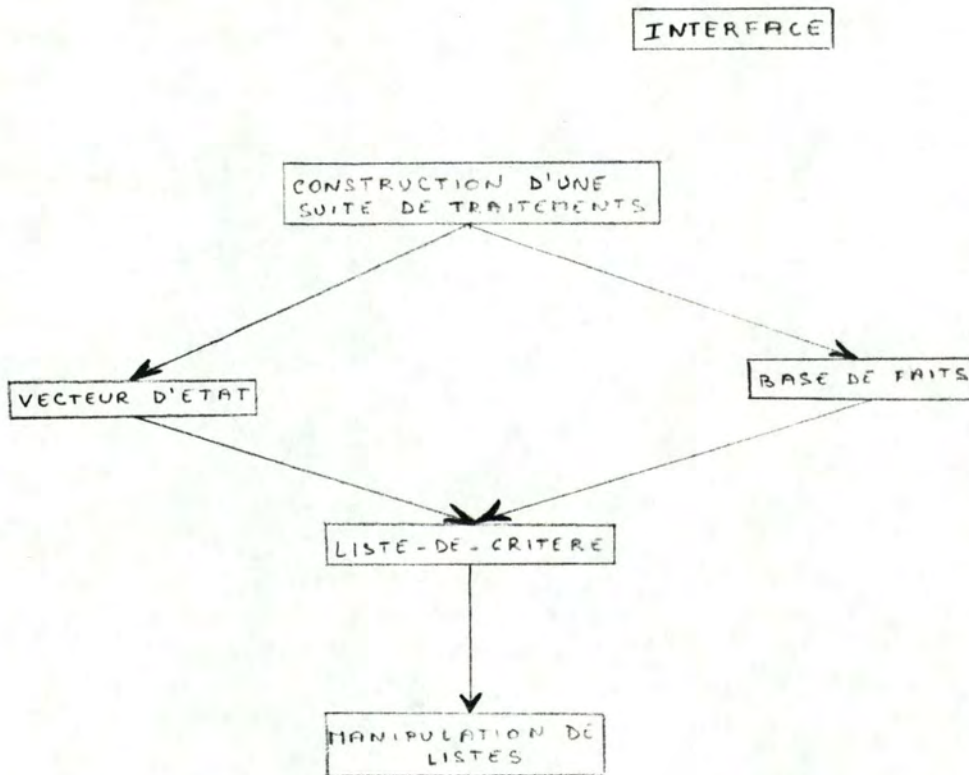
où L est une liste.

Sémantique : Chaque élément de L est la représentation interne d'un pigment composé de zinc.

4.4. IMPLEMENTATION

4.4.5. ARCHITECTURE DE LA BASE DE REGLES.

Nous avons d'abord écrit des primitives de bas niveau permettant la manipulation de listes et de LISTES DE CRITERE. Grâce à ces règles, nous avons pu implémenter les règles de manipulation des VECTEURS D'ETAT, celles de manipulation de la base de faits, celles du mécanisme de raisonnement et celles de l'interface. Nous allons dans ce paragraphe, présenter ces différentes classes de règles; la hiérarchie "utilise" qui les structure est la suivante.



4.4. IMPLEMENTATION

4.4.5.1. Procédures de manipulation de listes.

Cette classe de procédures se présente comme une boîte à outils que nous avons dû construire du fait que nous ne disposions que du langage PROLOG, dépourvu de tout environnement de programmation PROLOG.

Certaines procédures servent à déterminer l'appartenance d'un élément à une liste, l'inclusion d'une liste dans une autre, ou le résultat de la comparaison du rang de deux éléments dans une liste ordonnée. D'autres calculent le résultat de la concaténation de deux listes, de la différence entre deux listes ou de la suppression des répétitions d'éléments dans une liste.

Elles sont spécifiées comme suit :

- APPARTIENT(Element, Liste).

où Element est un atome,
Liste est une liste,

Cette procédure réussit ssi Element est un élément de la liste Liste.

- CONCAT(A, B, C).

où A, B, C sont des listes,

Cette procédure a pour effet d'instancier A au résultat de la concaténation de C à B.

- DEDOUBLE(A, B).

où A, B sont des listes,

Cette procédure a pour effet d'instancier A au résultat de la suppression des répétitions dans B.

- DIFF(A, B, C).

où A, B, C sont des listes,

Cette procédure a pour effet d'instancier C à une liste dont l éléments appartiennent à la différence entre l'ensemble des éléments de A et l'ensemble des éléments de B.

- INCLUS(A, B).

où A, B sont des listes,

Cette procédure réussit ssi l'ensemble des éléments de A est inclus dans l'ensemble des éléments de B.

4.4. IMPLEMENTATION

- INFEGAL(A, B, C).

où A, B sont des atomes,
C est une liste,

Cette procédure réussit ssi A et B appartiennent à C et si le rang de A dans la suite des éléments de C est inférieur ou égal à celui de B.

- INFEGALAU(A, B, C).

où A est un atome,
B, C sont des listes,

Cette procédure réussit ss'il existe au moins un élément de B, soit BB, tel que INFEGAL(A, BB, C) est établie.

4.4.5.2. Procédures de manipulation de LISTE DE CRITERE.

Ces procédures utilisent celles qui ont été définies dans le point précédent. Elles permettent de rechercher un CRITERE DE DESCRIPTION de NOM DE CRITERE donné dans une LISTE DE CRITERE ou de déterminer l'appartenance d'un CRITERE DE DESCRIPTION donné dans une LISTE DE CRITERE. Elles sont spécifiées comme suit :

- VALEUR_ITEM(Critere, Liste, Resultat).

où Critere est un CRITERE_DE_DESCRIPTION,
Liste est une LISTE_DE_CRITERE,
Resultat est un entier,

Cette procédure a pour effet d'instancier Resultat à :

- 1 si Critere appartient à Liste,
- 2 s'il n'existe pas C appartenant à Liste tel que
NOM_DE_CRITERE(C) = NOM_DE_CRITERE(Critere),
- 0 s'il existe C appartenant à Liste tel que
NOM_DE_CRITERE(C) = NOM_DE_CRITERE(Critere) et
(VALEUR(C) est différent de VALEUR(Critere)) ou
(LISTE_DE_VALEUR(C) est différent de LISTE_DE_VALEUR(Critere)).

- EXTVALCRIT(Nom, Liste, Valeur).

où Nom est un NOM_DE_CRITERE,
Liste est une LISTE_DE_CRITERE,
Valeur est une VALEUR ou une LISTE_DE_VALEUR.

Cette procédure a pour effet de rechercher dans Liste le CRITERE_DE_DESCRIPTION dont le NOM_DE_CRITERE est Nom et d'instancier Valeur à sa VALEUR_DE_CRITERE; si Nom n'est pas un NOM_DE_CRITERE dans Liste, Valeur est instanciée à LISTE_VIDE.

4.4. IMPLEMENTATION

4.4.5.3. Procédures de manipulation d'un VECTEUR D'ETAT.

Ces procédures utilisent les primitives de manipulation de LISTE DE CRITERE puisque, rappelons-le, le type d'objet VECTEUR D'ETAT est représenté à l'aide du type générique LISTE DE CRITERE. Elles ont pour rôle principal d'aller rechercher dans un VECTEUR D'ETAT un ITEM précisé. Leurs spécifications précises sont les suivantes :

- EXTNAT(Etat, Nature).

où Etat est une LISTE_DE_CRITERE,
Nature est une LISTE_DE_VALEUR.

Cette procédure a pour effet d'instancier Nature à la valeur de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est nature dans le VECTEUR D'ETAT dont la représentation interne est Etat. Si ce NOM DE CRITERE n'appartient pas au VECTEUR D'ETAT, Nature est instanciée à LISTE_VIDE.

- EXTFORME(Etat, Forme).

où Etat est une LISTE_DE_CRITERE,
Forme est une LISTE_DE_VALEUR.

Cette procédure a pour effet d'instancier Forme à la valeur de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est forme dans le VECTEUR D'ETAT dont la représentation interne est Etat. Si ce NOM DE CRITERE n'appartient pas au VECTEUR D'ETAT, Forme est instanciée à LISTE_VIDE.

- EXTENV(Etat, Env).

où Etat est une LISTE_DE_CRITERE,
Env est une LISTE_DE_VALEUR.

Cette procédure a pour effet d'instancier Env à la valeur de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est env dans le VECTEUR D'ETAT dont la représentation interne est ETAT. Si ce NOM DE CRITERE n'appartient pas au VECTEUR D'ETAT, Env est instanciée à LISTE_VIDE.

- EXTCARSPEC(Etat, Carspec).

où Etat est une LISTE_DE_CRITERE,
Carspec est une LISTE_DE_VALEUR.

Cette procédure a pour effet d'instancier Carspec à la valeur de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est carspec dans le VECTEUR D'ETAT dont la représentation interne est Etat. Si ce NOM DE CRITERE n'appartient pas au VECTEUR D'ETAT, Carspec est instanciée à LISTE_VIDE.

4.4. IMPLEMENTATION

- EXTETATSURF(Etat, Etatsurf).

où Etat est une LISTE_DE_CRITERE,
Etatsurf est une LISTE_DE_VALEUR.

Cette procédure a pour effet d'instancier Etatsurf à la valeur de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est etatsurf dans le VECTEUR D'ETAT dont la représentation interne est Etat. Si ce NOM DE CRITERE n'appartient pas au VECTEUR D'ETAT, Etatsurf est instanciée à LISTE_VIDE.

- EXTAS(Etat, Aspect).

où Etat est une LISTE_DE_CRITERE,
Aspect est une LISTE_DE_VALEUR.

Cette procédure a pour effet d'instancier Aspect à la valeur de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est aspect dans le VECTEUR D'ETAT dont la représentation interne est Etat. Si ce NOM DE CRITERE n'appartient pas au VECTEUR D'ETAT, Aspect est instanciée à LISTE_VIDE.

- EXTRES(Etat, Resist).

où Etat est une LISTE_DE_CRITERE,
Resist est une LISTE_DE_VALEUR.

Cette procédure a pour effet d'instancier Resist à la valeur de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est resist dans le VECTEUR D'ETAT dont la représentation interne est Etat. Si ce NOM DE CRITERE n'appartient pas au VECTEUR D'ETAT, Resist est instanciée à LISTE_VIDE.

- EXTPROT(Etat, Protect).

où Etat est une LISTE_DE_CRITERE,
Protect est une LISTE_DE_VALEUR.

Cette procédure a pour effet d'instancier Protect à la valeur de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est protect dans le VECTEUR D'ETAT dont la représentation interne est Etat. Si ce NOM DE CRITERE n'appartient pas au VECTEUR D'ETAT, Protect est instanciée à LISTE_VIDE.

- EXTDERNTRT(Etat, Dt).

où Etat est une LISTE_DE_CRITERE,
Dt est une LISTE_DE_VALEUR.

Cette procédure a pour effet d'instancier Dt à la valeur de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est derntrt dans le VECTEUR D'ETAT dont la représentation interne est Etat. Si ce NOM DE CRITERE n'appartient pas au VECTEUR D'ETAT, Dt est instanciée à LISTE_VIDE.

4.4. IMPLEMENTATION

- EXTITEMSVECT(Etat, N, F, E, C, ES, A, R, P, D).

où Etat est une LISTE_DE_CRITERE,
tous les autres arguments sont des LISTES_DE_VALEUR.

Cette procédure a pour effet d'instancier N, F, E, C, ES, A, R, P, D aux valeurs résultant de l'exécution des procédures Extnat(Etat, N), Extforme(Etat, F), Extenv(Etat, E), Extcarspec(Etat, C), Extetatsurf(Etat, ES), Extas(Etat, A), Extres(Etat, R), Extprot(Etat, P) et Extidenrtr(Etat, D).

4.4.5.4. Procédures de manipulation de la base de faits.

En plus des règles qui constituent la base de faits et qui sont directement utilisables par les procédures traduisant les mécanismes de construction d'un système de peinture, nous avons jugé utile de mettre au point des primitives capables de renvoyer des renseignements résultant d'un traitement des informations contenues dans la base de faits.

Il y a deux catégories de procédures de ce type : la première recherche un certain COMPOSANT d'une occurrence de TRAITEMENT, la seconde détermine les VALEURS d'une COMPOSANTE donnée qui appartient à une classe stipulée.

Elles sont spécifiées de la façon suivante :

a)- EXTPVC(Trt, Pvc).

où Trt est un NOM_DE_TRAITEMENT,
Pvc est un réel.

Cette procédure a pour effet d'instancier Pvc à la valeur du PVC du TRAITEMENT dont la représentation interne est Trt.

- EXTLIANT(Trt, Liant).

où Trt est un NOM_DE_TRAITEMENT,
Liant est un atome.

Cette procédure a pour effet d'instancier Liant à la valeur du LIANT du TRAITEMENT dont la représentation interne est Trt.

- EXTSOL(Trt, Solvant).

où Trt est un NOM_DE_TRAITEMENT,
Solvant est un atome.

Cette procédure a pour effet d'instancier Solvant à la valeur du SOLVANT du TRAITEMENT dont la représentation interne est Trt.

4.4. IMPLEMENTATION

- EXTPIG(Trt, Pigment).

où Trt est un NOM_DE_TRAITEMENT,
Pigment est une liste.

Cette procédure a pour effet d'instancier Pigment à la valeur du PIGMENT du TRAITEMENT dont la représentation interne est Trt.

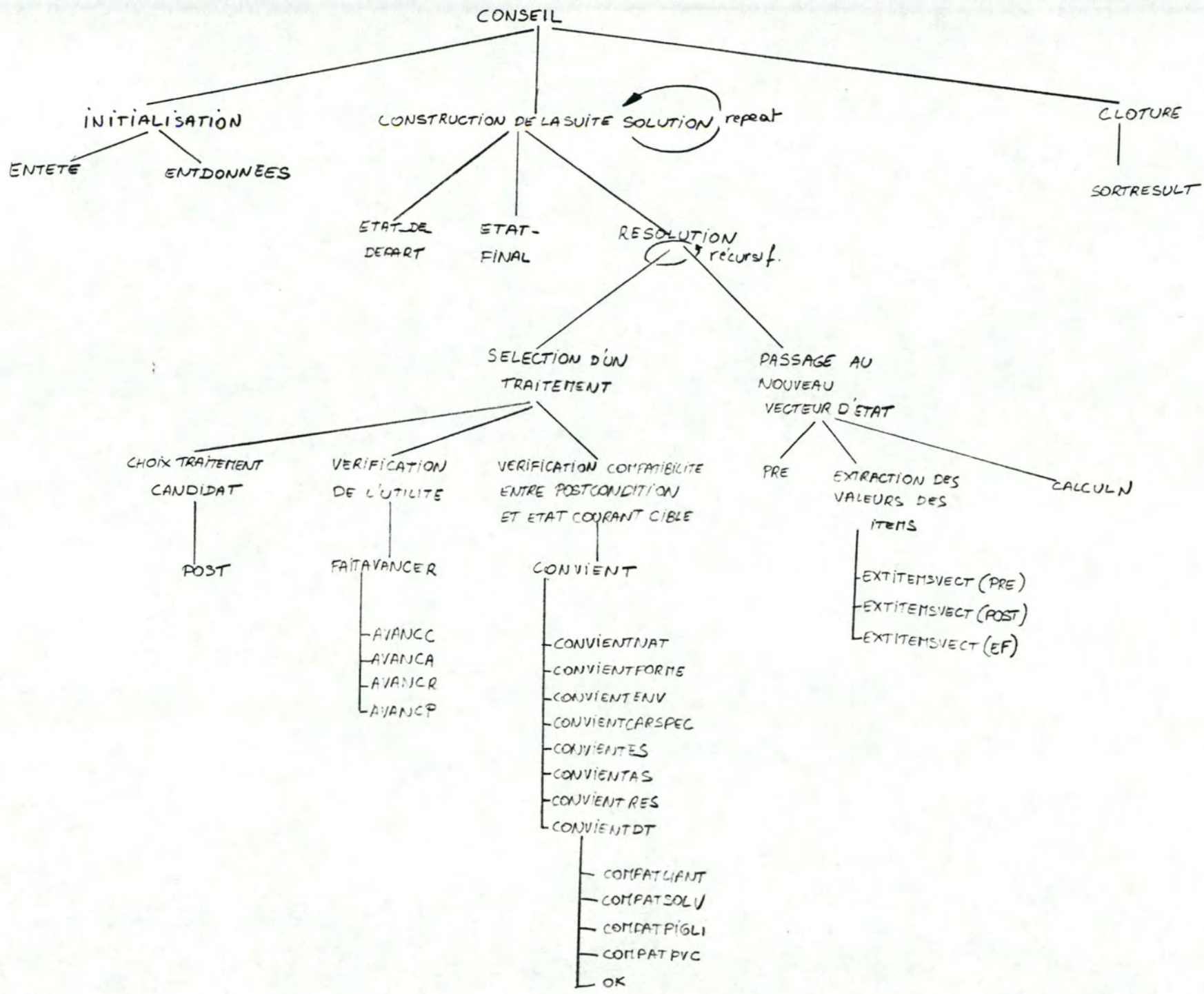
b)- EXTLISTITEM(Classe, Item, Liste, Res).

où Classe est un atome,
Item est un NOM_DE_CRITERE,
Liste et Res sont des LISTES_DE_VALEUR.

Cette procédure a pour effet d'instancier Res de telle sorte que les VALEURS DE CRITERE représentées par Res soient de la classe représentée par Classe et appartiennent à l'ensemble des VALEURS DE CRITERE représenté par Liste.

4.4.5.5. Procédures pour la construction d'une suite de traitements.

Ces règles constituent le coeur du programme puisqu'elles traduisent le raisonnement suivi dans le processus de conseil en peinture. Ce mode de raisonnement a été formalisé au point 4.3., et est décomposé en sous-tâches selon un ordre chronologique fixe. A ce niveau de l'architecture physique, l'arbre-ET de la tâche de conseil de mise en peinture peut être représenté par le schéma de la page suivante.



4.4. IMPLEMENTATION

4.4. IMPLEMENTATION

Certaines facilités du langage PROLOG et certaines méta-connaissances sur le domaine d'expertise permettent de modifier et d'accélérer la recherche exhaustive, en profondeur d'abord d'une solution. Les deux cas où nous avons observé de meilleurs résultats lors de l'usage de ces techniques sont les suivants :

- dans la sous-tâche "sélection du traitement", une des phases est la vérification de l'utilité de la sélection courante; si cet objectif échoue, grâce à l'introduction d'un "cut", le backtracking sera interrompu et la recherche s'orientera immédiatement vers un autre traitement-candidat;

- de la même façon, l'usage du "cut" lorsque la sous-tâche "convient" échoue, permet d'abandonner rapidement un traitement qui ne peut pas faire partie du système de peinture résultat, et de considérer un nouveau traitement-candidat.

L'utilisation de telles stratégies peut paraître relever de l'optimisation des performances lors de l'exécution, mais elle reflète assez bien l'approche d'experts humains construisant un système de peinture.

4.4.5.5.1. Sélection d'un TRAITEMENT.

a) Vérification de l'utilité de la sélection d'un TRAITEMENT.

- AVANCC(Cars, Carf).

où Cars et Carf sont des LISTES_DE_VALEUR.

Cette procédure réussit ssi les VALEURS DE CRITERE représentées par Carf sont également représentées par Cars.

- AVANCA(Ass, Asf).

où Ass et Asf sont des LISTES_DE_VALEUR.

Cette procédure réussit ssi au moins une des VALEURS DE CRITERE représentées par Ass est également représentée par Asf et fait partie de la classe remp ou supp.

- AVANCR(Ress, Resf).

où Ress et Resf sont des LISTES_DE_VALEUR.

Cette procédure réussit ssi dans la suite ordonnée des VALEURS DE CRITERE possibles pour le CRITERE DE DESCRIPTION dont le NOM DE CRITERE est resist, les VALEURS DE CRITERE représentées par Ress ont un rang supérieur ou égal à celui des VALEURS DE CRITERE représentées par Resf.

4.4. IMPLEMENTATION

- AVANCP(Pros, Prof).

où Pros et Prof sont des LISTES_DE_VALEUR.

Cette procédure réussit ssi au moins une des VALEURS DE CRITERE représentées par Prof est aussi représentée par Pros.

- FAITAVANCER(Cars, Carf, Ass, Asf, Ress, Resf, Pros, Prof).

où tous les arguments sont des LISTES_DE_VALEUR.

Cette procédure réussit ssi :

- une des procédures suivantes réussit :

 avancc(Cars, Carf) ou avanca(Ass, Asf) ou
 avancr(Ress, Resf) ou avancp(Pros, Prof)

ou bien si

- Cars est instanciée à la valeur [prepa] et si special(pas_prepa) échoue.

b) Vérification de la compatibilité entre l'ETAT COURANT et la POSTCONDITION d'un TRAITEMENT.

- CONVIENT (Trt, Es, Ef).

où Trt est un NOM_DE_TRAITEMENT,
Es et Ef sont des LISTES_DE_CRITERE.

Cette procédure réussit si les procédures suivantes réussissent. Chaque fois, le premier argument représente la valeur d'une COMPOSANTE du VECTEUR D'ETAT représenté par Ef et le second celle d'une COMPOSANTE du VECTEUR D'ETAT représenté par Es.

i. Compatibilité au niveau de la nature.

CONVIENTNAT(Natf, Nats).

où Natf et Nats sont des LISTES_DE_VALEUR.

Cette procédure réussit ssi :

- Natf = LISTE_VIDE, ou bien si
- Nats = LISTE_VIDE, ou bien si
- la VALEUR DE CRITERE représentée par Natf est également représentée par Nats, ou bien si
- membre(Nat, Nats) réussit, où Nat est .. une VALEUR_DE_CRITERE représentant la même VALEUR DE CRITERE que Natf.

4.4. IMPLEMENTATION

ii. Compatibilité au niveau de la forme.

CONVIENTFORME(Formef, Formes).

où Formef et Formes sont des LISTES_DE_VALEUR.

Cette procédure réussit ssi :

- Formef = LISTE_VIDE, ou bien si
- Formes = LISTE_VIDE, ou bien si
- dans la suite ordonnée des VALEURS DE CRITERE possibles pour le CRITERE DE DESCRIPTION dont le NOM DE CRITERE est forme, les VALEURS DE CRITERE représentées par Formes ont un rang strictement supérieur à celui des VALEURS DE CRITERE représentées par Formef.

iii. Compatibilité au niveau de l'environnement.

CONVIENTENV(Envf, Envs).

où Envf et Envs sont des LISTES_DE_VALEUR.

Cette procédure réussit ssi :

- Envf = LISTE_VIDE, ou bien si
- Envs = LISTE_VIDE, ou bien si
- dans la suite ordonnée des VALEURS DE CRITERE possibles pour le CRITERE DE DESCRIPTION dont le NOM DE CRITERE est env, les VALEURS DE CRITERE représentées par Envf ont un rang inférieur ou égal à celui des VALEURS DE CRITERE représentées par Envs.

iv. Compatibilité au niveau des caractéristiques spéciales.

CONVIENTCARSPEC(Carf, Cars).

où Carf et Cars sont des LISTES_DE_VALEUR.

Cette procédure réussit ssi :

- Carf = LISTE_VIDE, ou bien si
- toutes les VALEURS DE CRITERE représentées par Carf sont également représentées par Cars.

v. Compatibilité au niveau de l'état de surface.

CONVIENTES(Esf, Ess).

où Esf et Ess sont des LISTES_DE_VALEUR.

Cette procédure réussit ssi :

- toutes les VALEURS DE CRITERE représentées par Esf sont également représentées par Ess
- ou bien si
- equivalent(Esf, Ess) réussit.

4.4. IMPLEMENTATION

vi. Compatibilité au niveau de l'aspect.

CONVIENTAS(Asf, Ass).

où Asf et Ass sont des LISTES_DE_VALEUR.

Cette procédure réussit ssi toutes les VALEURS DE CRITERE de la classe perm ou remp représentées par Asf sont également représentées par Ass.

vii. Compatibilité au niveau des résistances.

CONVIENTRES(Resf, Ress).

où Resf et Ress sont des LISTES_DE_VALEUR.

Cette procédure réussit ssi :

- Resf = LISTE_VIDE, ou bien si
- dans la suite ordonnée des VALEURS DE CRITERE possibles pour le CRITERE DE DESCRIPTION dont le NOM DE CRITERE est resist, les VALEURS DE CRITERE représentées par Resf ont un rang inférieur ou égal à celui des VALEURS DE CRITERE représentées par Ress.

viii. Compatibilité au niveau chimique.

CONVIENTDT(Dtf, Dts).

où Dtf et Dts sont des LISTES_DE_VALEUR.

Cette procédure réussit ssi :

- Dtf = LISTE_VIDE, ou bien si
- Dts = LISTE_VIDE, ou bien si
- les procédures suivantes réussissent : compatliant(F, S), compatsolv(F, S), compatpigli(F, S) et compatpvc(F, S) où F (resp. S) est une VALEUR_DE_CRITERE représentant la même VALEUR DE CRITERE que Dtf (resp. S).

- COMPATLIANT(F, S).

où F et S sont des NOMS_DE_TRAITEMENT.

Cette procédure réussit ssi dans l'échelle de thermoplasticité des liants, le liant du TRAITEMENT représenté par F a un coefficient inférieur ou égal à celui du liant du TRAITEMENT représenté par S.

4.4. IMPLEMENTATION

- COMPATSOLV(F, S).

où F et S sont des NOMS_DE_TRAITEMENT.

Cette procédure réussit ssi dans l'échelle d'agressivité des solvants, le solvant du TRAITEMENT représenté par F a un coefficient inférieur ou égal à celui du solvant du TRAITEMENT représenté par S.

- COMPATPIGLI(F, S).

où F et S sont des NOMS_DE_TRAITEMENT.

Cette procédure réussit ssi ok(Ps, Lf) réussit (avec Ps représentant les pigments du TRAITEMENT représenté par S et Lf représentant le liant du TRAITEMENT représenté par F).

- COMPATPVC(F, S).

où F et S sont des NOMS_DE_TRAITEMENT.

Cette procédure réussit ssi le pvc du TRAITEMENT représenté par F est inférieur ou égal à celui du TRAITEMENT représenté par S.

- OK(P, L).

où P est une liste et
L est un atome.

Cette procédure réussit ssi :

- L représente un liant non-saponifiable, ou bien si
- P = LISTE_VIDE, ou bien si
- les pigments représentés par P ne sont pas composés de zinc.

4.4.5.5.2. Calcul du nouveau VECTEUR D'ETAT.

- CALCULN(Ec,
Ass, Ress, Pros,
Carp, Esp, Asp, Resp, Prop, Dtp,
Asf, Resf, Prof).

où Ec est une LISTE_DE_CRITERE et
tous les autres arguments sont des LISTES_DE_VALEUR.

Cette procédure a pour effet d'instancier Ec à la valeur d'un VECTEUR D'ETAT dont :

"aspect"	a pour valeur celle représentée par	Asf + Asp + (Asf-Ass)
"resist"	"	Resf + Resp + (Resf-Ress)
"protect"	"	Prop + (Prof - Pros)
"carspec"	"	Carp
"etatsurf"	"	Esp
"derntrt"	"	Dtp

4.4. IMPLEMENTATION

4.4.5.6. Interface.

4.4.5.6.1. Primitives de programmation de l'interface.

1. Inscription d'une phrase à l'écran.

- SORTPHR(Phrase)

où Phrase est une liste d'atomes.

Cette procédure a pour effet de sortir à l'écran la phrase composée des mots contenus dans Phrase. Si cette liste est vide, la règle a pour effet de faire passer le curseur à la ligne suivante.

2. Inversion de l'ordre des éléments d'une liste.

- RENUListe(Liste_A_Renverser, Liste_Auxiliaire,
Liste_Renversée).

où Liste_A_Renverser, Liste_Auxiliaire et
Liste_Renversée sont des listes.

Cette procédure a pour effet d'inverser l'ordre des éléments de Liste_A_Renverser à l'aide de Liste_Auxiliaire et d'instancier Liste_Renversée au résultat de cette opération.

3. Sortie d'une liste à l'écran.

- SORTListe(Liste).

où Liste est une liste.

Cette procédure a pour effet de sortir à l'écran les éléments de la liste Liste en passant à une nouvelle ligne après chacun d'eux.

4. Construction d'un ETAT DE SURFACE.

- ACCOUPLE(Cor, Gr, Pous, Prop, Acc, Net,
[C, G, Po, Pr, A, N]).

où Cor, Gr, Pous, Prop, Acc et Net sont des mots,
[C, G, Po, Pr, A, N] est une LISTE_DE_VALEUR.

Cette procédure a pour effet de construire la VALEUR DE CRITERE [C, G, Po, Pr, A, N] du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est etatsurf à partir des mots Cor, Gr, Pous, Prop, Acc et Net.

4.4. IMPLEMENTATION

5. Couplage des valeurs.

a. Couplage de la corrosion.

- COUPLEC(Cor, C).

où Cor est un mot
C est une VALEUR.

Cette procédure a pour effet d'instancier la VALEUR C qui correspond à la corrosion de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est etatsurf, à la valeur correspondant au mot Cor.

b. Couplage de la graisse.

- COUPLEG(Gr, G).

où Gr est un mot
G est une VALEUR.

Cette procédure a pour effet d'instancier la VALEUR C qui correspond à la graisse de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est etatsurf, à la valeur correspondant au mot Gr.

c. Couplage de la poussière.

- COUPLEPO(Pous, Po).

où Pous est un mot
Po est une VALEUR.

Cette procédure a pour effet d'instancier la VALEUR C qui correspond à la poussière de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est etatsurf, à la valeur correspondant au mot Pous.

d. Couplage de la propreté.

- COUPLEPR(Prop, Pr).

où Prop est un mot
Pr est une VALEUR.

Cette procédure a pour effet d'instancier la VALEUR C qui correspond à la propreté de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est propreté, à la valeur correspondant au mot Prop.

4.4. IMPLEMENTATION

e. Couplage de l'accrochage.

- COUPLEA(Acc, A).

où Acc est un mot
A est une VALEUR.

Cette procédure a pour effet d'instancier la VALEUR C qui correspond à l'accrochage de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est etatsurf, à la valeur correspondant au mot Acc.

f. Couplage de la netteté.

- COUPLEN(Net, N).

où Net est un mot
N est une VALEUR.

Cette procédure a pour effet d'instancier la VALEUR C qui correspond à la netteté de la VALEUR DE CRITERE du CRITERE DE DESCRIPTION dont le NOM DE CRITERE est etatsurf, à la valeur correspondant au mot Net.

4.4.5.6.2. Interface proprement dite.

1. Entête du programme.

- ENTETE.

Cette procédure a pour effet d'afficher à l'écran le texte introductif suivant :

"Bonjour. Veuillez entrer successivement
l'état initial et l'état désiré du support
après traitement."

2. Entrée des données.

a. Entrée de l'état initial.

- ENTETATINIT(Nature, Forme, Environnement).

où Nature, Forme et Environnement sont des LISTES_DE_VALEUR.

Cette procédure a pour effet de lire au clavier les valeurs des critères d'état de l'état initial, et d'ajouter à la base une règle ETAT_DE_DEPART(E) construite à partir de ces valeurs. En particulier, Nature, Forme et Environnement seront instanciées à la valeur de la VALEUR DE CRITERE des CRITERES DE DESCRIPTION

4.4. IMPLEMENTATION

dont les NOMs DE CRITERE sont respectivement Nature, Forme et Environnement.

b. Entrée de l'état final.

- ENTETATFIN(Nature, Forme, Environnement).

où Nature, Forme et Environnement sont des LISTEs_DE_VALEUR.

Cette procédure a pour effet de lire au clavier les VALEURs DE CRITERE des CRITERES DE DESCRIPTION de l'état final, excepté la nature, la forme et l'environnement, et d'ajouter à la base une règle ETAT_FINAL(E) construite à partir des valeurs lues au clavier et des valeurs Nature, Forme et Environnement.

c. Entrée des deux états final et initial.

- ENTDONNEES.

Cette procédure a pour effet de faire se succéder l'entrée de l'état initial et de l'état final.

5. CONCLUSION

5. CONCLUSION.

L'évaluation de notre système est une question épineuse. En effet, les attentes (et espoirs !) vis-à-vis des performances d'un système expert sont souvent hasardeuses. En principe, le programme doit s'exécuter en un temps raisonnable et être capable de traiter un ensemble de cas "typiques", peut-être ceux-là mêmes utilisés pour obtenir la connaissance du domaine d'expertise. On peut espérer que l'on mettra sous peu, au point une série de critères permettant d'établir de façon fiable qu'un système expert a atteint un certain niveau de performance.

Dans des domaines réels, tels que le nôtre, où le manque de systématisation mathématique ou logique est évident, il est peu probable que l'on puisse un jour "prouver" qu'un système expert est effectivement un expert au sens propre du terme. L'argumentation est empirique, et une fois encore, le nombre de situations que le programme pourra rencontrer sera très vaste dans toute application non-triviale.

De plus, la technique qui consiste à comparer les performances d'un système expert et d'un expert humain n'est pas facile à mettre en oeuvre. En effet, il faut garder en mémoire que la comparaison est difficile puisqu'un expert humain dispose pour donner ses solutions d'autres moyens qu'un système automatisé; on peut citer par exemple la collaboration avec des collègues, l'expérimentation, la consultation d'archives et des mécanismes de raisonnement plus puissants que le chaînage, tels que la généralisation, le raisonnement par analogie, etc.

On peut se demander ce qui reste comme moyen d'évaluation puisque ni les outils mathématiques, ni l'expérimentation ne sont concluants. On pourrait considérer les faibles notions d'adéquation, de parcimonie et d'élégance.

Un système est adéquat à une tâche s'il résout avec succès des jeux de test construits selon une méthode du type couverture de classes de cas [VAN84], et si on peut expliquer ses éventuelles erreurs.

Un système est parcimonieux s'il n'utilise pas plus de moyens conceptuels et matériels que nécessaire.

Un système est élégant s'il montre une certaine variété dans ses méthodes; il possède une méthode de résolution par classe de problèmes et des interfaces clairs entre les différents modules.

Ces critères sont présentés dans [JAC85] et font référence à une discussion à propos de modèles psychologiques, ce qui tendrait à montrer que les critères d'évaluation des produits de l'ingénierie de la connaissance ne sont pas très différents des critères de jugement des théories psychologiques basées sur des métaphores informatiques.

Nous pensons que notre système est adéquat à la tâche qu'il exécute. Bien que nous n'ayons pas eu le temps de construire des jeux de test rigoureux reprenant toutes les classes de cas possibles et

5. CONCLUSION

leurs limites, nous avons testé le système sur un ensemble de cas variés, qui reprenait des cas limites, c'est-à-dire des cas pour lesquels les données sont soit incomplètes, soit contradictoires.

Quant à la parcimonie, la base de connaissances ne reprend que des règles qui sont utiles au fonctionnement du système. Par contre, leur représentation n'est pas la plus économique : il fallait choisir entre un programme peu gourmand en place mémoire et illisible, et un programme clair mais plus gourmand. Nous avons opté pour la deuxième solution, en vue des extensions qui seront réalisées par d'autres que nous.

De même, la clarté et l'uniformité des traitements dans le programme doit permettre de réaliser facilement des extensions ultérieures. En effet, les règles de choix d'un traitement sont relativement simples puisque pour un critère d'un traitement, il existe une règle de sélection et une règle de modification du support. De plus, l'architecture logicielle est basée sur une hiérarchie "utilise". En d'autres termes, nous avons exploité la possibilité de "cacher de l'information" dans des modules qui sont "utilisés" par des modules de niveau supérieur. Ces modules consistent en des primitives paramétrées qui manipulent des types d'objets et des types génériques. Dès lors, ces modules sont réutilisables pour des extensions futures du système.

5.1. PERFORMANCES.

L'utilisation du langage est assez coûteuse au point de vue de la machine. En effet, le nombre de traitements de listes impliqués par la reconnaissance de ce qu'une liste est l'instance d'une autre est fonction de la longueur et du nombre de listes imbriquées, et du nombre de variables dans le pattern. Le mécanisme de backtracking est également coûteux, car il suppose une gestion de pointeurs assez complexe : tous les objectifs qui sont en cours de preuve reçoivent un pointeur, de façon à ce que l'interpréteur puisse les retrouver aisément au moment où il doit "défaire" les instanciations déjà établies.

De plus, nous n'avons pas implémenté de méta-règles qui pourraient orienter le cours de la recherche vers les "bonnes" règles. Ainsi, le processus de recherche dans la base de connaissances est aveugle, et le parcours de la base de connaissances se fait toujours de haut en bas à travers les règles, et de gauche à droite parmi les sous-objectifs d'une règle.

Il est certain que l'utilisation d'un interpréteur PROLOG n'est pas l'idéal au point de vue des performances. En effet, la technique de recherche exhaustive en profondeur d'abord est très coûteuse et peut ne jamais se terminer. De plus, le backtracking automatique cause parfois des difficultés lorsqu'on examine les traces de l'exécution du programme, et se révèle également très coûteux. Cependant, l'utilisation de ces techniques améliore la rapidité et la facilité de la construction d'un prototype, tout en lui conférant une grande lisibilité.

5. CONCLUSION

En plus de ces problèmes généraux, nous pouvons expliquer les mauvaises performances de notre prototype au niveau de la vitesse d'exécution par l'absence de toute optimisation d'une part et par le caractère verbeux du texte du programme. Ce caractère résulte du choix que nous avons opéré pour faciliter la lisibilité, la modifiabilité et les tests de la base de connaissances. Nous aurions pu, par exemple, réduire au strict minimum les chaînes de caractères impliquées dans les traitements de listes. Encore une fois, c'est un objectif de lisibilité qui nous a fait rejeter cette option.

5.2. EXTENSIONS.

Les constructeurs de RI ont fait certaines constatations sur le comportement des experts lors de la construction de leur système. Ces constatations se révèlent également vraies dans notre cas :

- les experts humains ont une idée claire de la décomposition de la tâche principale en sous-tâches et des relations chronologiques entre ces sous-tâches;

- à l'intérieur des sous-tâches, leur comportement est davantage guidé par les exceptions que par les cas généraux;

- les experts humains éprouvent des difficultés lorsqu'ils doivent se souvenir à la demande de circonstances exceptionnelles; une fois encore, ils sont guidés par les événements.

Comme nous l'avons déjà souligné dans le troisième chapitre, l'idéal est de procéder cas par cas, c'est-à-dire de se présenter chez les experts avec un guide d'interview en se ménageant un maximum de chances pour des explications ouvertes. Nous pensons que c'est un problème de relations humaines, à savoir un processus relevant de la psychologie et difficilement formalisable. En tous cas, un des atouts du succès de la construction d'un système expert est l'intérêt des experts humains eux-mêmes pour le projet.

5.3. VOIES D'AVENIR.

Il est possible d'étendre le sous-système dans plusieurs sens différents.

Le premier est de traiter tous les types de supports non-métalliques. Les autres types à intégrer sont les bois, les maçonneries et les supports de type particulier tels les matières plastiques, le verre, la faïence, etc. Le processus de traitements des supports d'autre nature que le métal est le même que celui appliqué aux métaux. Il suffit d'adapter et de compléter la terminologie employée aux spécificités des autres natures. Les jeux de valeurs des caractéristiques sont évidemment différents (on ne trouve pas de rouille sur une maçonnerie, ni de champignons sur un métal!), mais le processus de raisonnement global reste le même. En fonction des options que nous avons choisies pour représenter la connaissances (objets et types d'objets génériques), les outils que nous avons développés doivent pouvoir être réemployés pour cette extension.

5. CONCLUSION

Le deuxième est de traiter l'ensemble des produits offerts par la firme, et qui ne sont pas d'un usage courant, tels les produits conçus pour les industries ou pour des cas tout à fait spécifiques (produits pour piscines ou pour bateaux, par exemple). Il suffira d'ajouter à la base de connaissances les nouveaux produits, et de décrire leurs usages suivant la formalisation et l'implémentation que nous avons employées.

Un autre encore, serait d'étendre le système en y intégrant des connaissances inexactes ou incomplètes ou encore avec des coefficients de préférence. À vrai dire, le système ne sera vraiment expert que lorsqu'il pourra "raisonner" dans un tel univers "incomplet". Un exemple d'extension de ce type est celui de l'introduction d'un filtre commercial autour du noyau technique.

Le plus important est la convivialité avec l'utilisateur. Ainsi, il faudra créer une interface entre l'utilisateur et le système expert, qui lui permettra de donner ses renseignements d'une façon agréable, et de recevoir la réponse à sa ou ses questions sous forme lisible. Rappelons ici que ce type d'interface fait toujours partie d'un système expert "bien conçu", et qu'il est important que l'utilisateur, qui n'est pas toujours un informaticien, ne se sente pas perdu devant les notations propres à l'informatique. Dès lors, cette interface doit faire appel à un sous-ensemble du langage naturel.

De plus, une interface d'acquisition de nouvelles connaissances et d'explication du mécanisme d'obtention du résultat devraient faire partie d'un système expert à son stade final. Il faut penser que ces interfaces ne doivent pas être employées par des informaticiens, et qu'elles doivent être très conviviales.

Nous proposerons aux personnes qui seraient amenées à prolonger ce travail de tenir compte des éléments suivants :

- nous avons signalé qu'une explication de la lenteur d'exécution est le caractère verbeux du code du programme mais n'oublions pas que la base de faits est loin d'être complète et donc que les performances ne peuvent que se dégrader si l'implémentation n'est pas optimisée;
- un point tout aussi important concerne la vérification à faire des critères de description des traitements et supports retenus dans le prototype et sans doute l'augmentation du nombre de ces critères.
- enfin, les règles de sélection d'un traitement et de modification de l'état d'un support devront être affinées en tenant compte précisément des modifications apportées aux critères de description.

BIBLIOGRAPHIE.

- [BAR82] BARR, A. and FEIGENBAUM, E.A. : The Handbook of Artificial Intelligence, tome 2, (HeurisTech Press, Stanford, California), 1982.
- [BOD83] BODART, F. et PIGNEUR, Y. : Conception assistée des applications informatiques; tome 1 : étude d'opportunité et analyse conceptuelle, (Masson, Presses Universitaires de Namur), 1983.
- [BON84] BONNET, Alain : L'intelligence artificielle : Promesses et Réalités, (InterEditions, Paris), 1984.
- [BUC84] BUCHANAN, B.G., and SHORTLIFFE, E.H. : Rule-Based Expert System The MYCIN experiments of the Stanford Heuristic Programming Pro (Addison - Wesley), 1984.
- [CLA84] CLANCEY, W. and SHORTLIFFE, E. : Readings in Medical Artificial Intelligence : The First Decade, (Addison-Wesley), 1984.
- [CLO84] CLOCKSIN, W.F. and MELLISH, C.S. : Programming in Prolog, (Springer - Verslag, Berlier Heidelberg), seconde éd. , 1984.
- [FEI84] FEIGENBAUM, E.A., and McCORDUCK, P. : La cinquième génération, (InterEditions, Paris), 1984.
- [FIK71] FIKES, R. and NILSSON, N. : STRIPS : a new approach to the application of theorem proving to problem solving, in Artificial Intelligence, 2 (3/4), pp 189 - 208, (North-Holland), 1971.
- [GIA85] GIANNESINI, F., KANOUI, H., PASERO, R. and VAN CANEGHEM, M. : Prolog, (InterEditions, Paris), 1985.
- [HAM85] HAMMOND, P., KRIWACZEK, F., SERGEOT, M. : Logic programming for expert systems, Professorship in International Computer Science, EXPERT SYSTEMS Mons - Namur, 1985.
- [HAY83] HAYES - ROTH, F., WATERMAN, D.A. and LENAT, D.B. : Building Expert Systems, (Addison - Wesley), 1983.
- [HAY84] HAYES - ROTH, F. : The Knowledge-Based Expert Systems : A Tutorial, in COMPUTER, pp 11 - 28, (IEEE), September 1984.
- [JACK85] JACKSON, P. : Knowledge Representation, Professorship in International Computer Science, EXPERT SYSTEMS Mons - Namur, 1985.
- [KAR84] KARNA, K.N. : Artificial Intelligence for man-machine interface in COMPUTER, pp 8 - 9, (IEEE), September 1984.

BIBLIOGRAPHIE

- [KLU85] KLUZNIAK, Feliks and SPAKOVICZ, Stanislaw with contribution by BIEN, S. Janusz : Prolog for Programmers, (Academic Press), 1985
APIC Studies in Data Processing n° 24.
Institute of Informatics, Warsaw University, Warsaw, Poland.
- [LAU82] LAURIERE, J.-L. : Représentation et utilisation des connaissances
in Technique et Science Informatiques(TSI), Vol. 1, n°s 1 et 2,
Janvier à avril 1982.
- [NIL71] NILSSON, NILS, J. : Problem-solving methods in Artificial Intelligence, (McGRAW-HILL), 1971.
- [ROB68] ROBINSON, J.A. : The Generalized resolution principle,
in Machine Intelligence 3, D. Mitchie Editor, pp 77-94,
(Edinburgh University Press), 1968.
- [ROL82] ROLLAND, C. et RICHARD, C. : The REMORA Methodology for information system design, in Information System Design Methodology-a comparative review, pp 369 - 426,
(North-Holland), 1982.
- [STE82] STEFIK, M., AIKINS, J., BALZER, R. and al. :
The Organization of Expert Systems. A Tutorial,
in Artificial Intelligence, 18, pp 135 - 173, (North-Holland),

BIBLIOGRAPHIE TECHNIQUE.

- [OCC83] OCCAA : Surface Coatings,
OCCAA : Oil and Colour Chemist's Association - Australia
(Chapmann and Hall), 1983 (2 tomes).
- [NYL85] NYLEN, Paul and SUNDERLAND Edward : Modern Surface Coatings,
(London, Interscience Publishers, a division of John Wiley a Sons Ltd), 1965.
- [LAM82] LAMBERT : Guide des systèmes anti-corrosion,
(Dpt Anti-corrosion de Trimetal FRANCE), Mai 1982.
- [SVE62] SVERIGES STANDARDSERINGSKOMMISSION : Pictorial Surface Preparations Standards for Painting Steel Structures, 1962.
(Svensk Standard SIS 05-59-00), 1962.

Guide Trimétal : Peintures Faciles,
(Trimétal-Belgique).

Guide Trimétal : Documentation pour peintres professionnels,
(Trimétal-Belgique).

BIBLIOGRAPHIE

Guide Trimetal : Fiches techniques des peintures,
(Trimétal-Belgique).

Cartes Trimetal : "Travaux de Peinture sur Métaux",
"Travaux de Peinture sur Bois et Dérivés",
"Travaux de Peinture sur Maçonneries, Béton,
Ciment, Plâtre et Dérivés",
(Trimétal-Belgique).

Conversations avec les experts en chimie et en conseil peinture

ANNEXES

ANNEXE A

ANNEXE TECHNIQUE.

PLAN.

1. Introduction.
2. Qu'est-ce qu'une peinture ?
 - 2.1. la composition chimique.
 - 2.1.1. les pigments.
 - 2.1.2. les solvants.
 - 2.1.3. les liants.
 - 2.2. le P.V.C. (Pigment Volume Concentration).
 - 2.3. la thermoplasticité, la dureté et l'adhérence.
3. Les systèmes de peinture.
 - 3.1. construction d'un système de peinture.
 - 3.2. caractérisation des diverses couches du système.
 - 3.2.1. les préparations et enduisages.
 - 3.2.2. les primers.
 - 3.2.3. les surfacers.
 - 3.2.4. les finitions.
 - 3.2.5. remarque.
 - 3.3. problématique des compatibilités.
 - 3.4. exemple de fiche de système de peinture.
4. Le conseil en peinture.
 - 4.1. critères.
 - 4.2. comment s'en tirer pratiquement ?
 - 4.2.1. caractérisation de l'objet à peindre.
 - 4.2.2. caractérisation des exigences du peintre.
 - 4.2.3. adoption d'un système satisfaisant les points 4.2.1 et 4.2.2 .

ANNEXE A

1. INTRODUCTION.

Le but de cette annexe est d'introduire le lecteur à la terminologie habituellement employée dans le domaine de la peinture en bâtiments, et aux concepts techniques de ce domaine.

Le lecteur pourra s'y référer si la signification d'un terme lui échappe.

Elle décrit également les principes mis en oeuvre pour déterminer et conseiller un système de peinture visant à atteindre l'état décrit comme optimal par l'utilisateur, étant donné un état initial préalablement spécifié.

Le lecteur trouvera la bibliographie de cette annexe en dernière partie de la bibliographie générale.

2. QU'EST-CE QU'UNE PEINTURE ?

Une peinture est une substance chimique composée de trois groupes d'éléments principaux : les pigments, les solvants et les liants. Chacun de ces composants possède un rôle propre et des caractéristiques qu'il transfère au produit composé.

En outre, le dosage de ces éléments permet de dériver trois autres propriétés, qui sont le P.V.C (Pigment Volume Concentration), la thermoplasticité et la dureté. On peut déterminer à partir des deux dernières propriétés l'adhérence du produit suivant certaines règles de compatibilité décrites au point 3.3.1 .

2.1. LA COMPOSITION CHIMIQUE.

Les trois composants majeurs peuvent être des groupes d'éléments. Dans ce cas, c'est soit l'élément qui se trouve en plus grande quantité, soit la réaction chimique des éléments entre eux qui donne ses caractéristiques à la peinture.

2.1.1. Les pigments.

Il existe deux classes de pigments : les pigments et les matières tinctoriales. Les pigments offrent un degré plus ou moins grand d'opacité, tandis que les matières tinctoriales fournissent un pouvoir colorant tout en laissant le produit translucide. Ces dernières sont donc employées dans la composition des vernis et des lasures (lasure : produit qui teint le bois sans laisser d'épaisseur sur le support, c'est-à-dire produit, teintant ou non, non-filmogène).

Par la suite, nous emploierons le terme général "pigment" pour les deux classes de produits, leur différenciation n'étant pas pertinente pour ce qui va suivre.

Les pigments sont utilisés en peinture pour remplir une fonction optique et/ou protectrice.

La fonction optique est la plus importante des caractéristiques d'un pigment. En effet, l'indice de réfraction du mélange pigment - liant détermine la couleur, l'opacité et la brillance du produit. La forme des pigments, la distribution de leur taille et leur dispersion dans le liant interviennent également : elles déterminent le grain de la peinture, l'épaisseur de la couche appliquée sur le support, et son aspect lisse ou structuré.

La fonction protectrice est moins apparente, mais tout aussi importante. Il s'agit ici de caractéristiques telles que la dureté de la surface, la résistance du produit aux intempéries et aux rayons solaires, sa flexibilité, son pouvoir anti-corrosion, son pouvoir de tuer les larves des insectes xylophages et d'empêcher de nouvelles pontes, et son adhérence. Cette fonction résulte en partie du pouvoir que possède le pigment de réagir avec le support avec lequel il entre

ANNEXE A

en contact, en particulier pour les pigments anti-corrosion.

Chaque pigment possède des fonctions protectrice et optique propres, qu'ils confèrent à la peinture dans la composition de laquelle ils entrent.

2.1.2. Les solvants.

Un solvant peut être défini comme une substance capable d'amener, par mélange, un solide à l'état fluide. Il est destiné à rendre possible l'application d'un composé pigment - liant sur un support et à s'évaporer ensuite.

Les solvants sont caractérisés par leur taux d'évaporation, c'est-à-dire la proportion de solvant évaporée par unité de temps. Ce taux peut être modifié par l'humidité et la chaleur de l'air ambiant.

Du taux d'évaporation, on peut dériver le temps et les conditions de séchage d'un produit. Si plusieurs solvants entrent dans la composition d'un même produit, certains s'évaporant plus vite que d'autres, il faut être certain que l'évaporation du mélange de solvants est compatible avec le mélange pigment - liant.

La force et l'efficacité de tous les solvants doivent être contrôlées conformément à des normes très strictes. Ainsi, des tests sont menés pour déterminer leur viscosité, leur dilution, leur pouvoir solvant, leur pureté, la température à laquelle ils peuvent exploser dans un récipient fermé en raison des vapeurs qu'ils dégagent (c'est-à-dire le flash point, utile pour le transport et la conservation des produits), et enfin leur toxicité.

2.1.3. Les liants.

Les liants, ou résines, servent, comme leur nom l'indique, à lier les pigments entre eux. Ils ne jouent pas de grand rôle individuellement : c'est leur mélange avec les pigments qui donne à une peinture ses caractéristiques.

Par contre, il faut en tenir compte dans la construction d'un système de peinture, en raison des incompatibilités potentielles qui peuvent survenir, et qui seront décrites au point 3.3.1 .

ANNEXE A

2.2. le P.V.C. (Pigment Volume Concentration).

Le P.V.C. est défini comme le pourcentage en volume de pigments dans le volume total solide de la peinture. L'effet du P.V.C. se mesure sur trois propriétés importantes : l'opacité, la durabilité et la brillance. Ces trois propriétés varient conflictuellement : la brillance maximale correspond à un P.V.C. très bas, la durabilité maximale à un P.V.C. bas et l'opacité maximale à un P.V.C. haut. De plus, Le P.V.C. affecte la flexibilité du film, son adhérence au support, sa résistance aux chocs, sa perméabilité à l'humidité et son farinage aux rayons UV.

Les variations de ces trois facteurs en fonction du P.V.C. sont décrites par le graphique suivant. Il n'est tenu compte pour celui-ci d'aucune échelle. Seule compte la forme des courbes.

ANNEXE A

2.3. La thermoplasticité, la dureté et l'adhérence.

La thermoplasticité peut être quantifiée par la capacité du film de peinture à suivre les mouvements du support dus à la chaleur. Un film très thermoplastique suivra parfaitement ces mouvements. Il est donc plus mou qu'un film non-thermoplastique (dit aussi duroplaste). La thermoplasticité peut servir de mesure de dureté de la peinture.

L'adhérence peut être exprimée sous forme d'une résultante de deux forces. La première est la force qui lie la couche de peinture à son support. La deuxième est la force de cohésion des particules d'un produit, et s'appelle "force de Vander Waals". Ces deux forces s'exercent dans des sens perpendiculaires. Cette situation peut être représentée par le schéma ci-dessous.

La première des deux forces décrites au paragraphe précédent est une force chimique, qui soit provient de la réaction chimique du produit avec le support, soit est une application de la loi des forces d'attraction de deux corps. En particulier elle dépend du rapport entre la porosité du support et la grosseur des pigments de la peinture. En effet, l'attraction exercée par un corps sur un autre est inversement proportionnelle au carré de leur distance. Plus loin est le pigment de la surface, moins grande est l'attraction que cette surface exerce sur lui. Les deux schémas qui suivent éclairent l'application de cette loi en peinture, et son effet sur l'adhérence du produit.

3. LES SYSTEMES DE PEINTURE.

Un système de peinture est une séquence d'une ou plusieurs opérations à effectuer sur le support donné en vue de le faire répondre aux exigences de l'utilisateur. Il est composé de quatre types d'opérations : les préparations et enduisages, les primers, les surfacers et les finitions.

3.1. CARACTERISATION DES DIVERSES COUCHES DU SYSTEME.

3.1.1. Les préparations et enduisages.

Les préparations et enduisages sont des traitements destinés à corriger l'état initial du support afin de lui permettre d'être recouvert de peinture. En effet, pour pouvoir être recouvert de peinture, un support ne doit pas présenter d'irrégularités de surface, il doit être propre, dépoussiéré et dégraissé, ne doit pas être corrodé; les trous du bois doivent être bouchés, les noeuds enlevés; les ciments et maçonneries ne doivent pas fariner, et enfin, tout support déjà peint et à rafraichir doit être dégraissé, et non-écaillé. Cette liste de dégradations n'est pas exhaustive. De plus, les dégradations sont souvent spécifiques à un type de support, et tous les supports envisageables ne sont pas cités ci-dessus.

Nous appelons "préparation" toute opération de ce type qui ne nécessite pas de produits de la firme, et "enduisage" toute opération qui en nécessite un.

3.1.2. Les primers.

Le primer est, comme son nom l'indique, la première couche à étendre sur le support préalablement enduit et préparé. Il a plusieurs rôles. L'un d'eux est d'opacifier le fond, c'est-à-dire faire en sorte que la couleur du fond ne passe pas à travers la finition. Un autre est de constituer une barrière entre le support et les autres couches de peinture, et, en particulier, entre le support et le milieu ambiant. Etant appliqué directement sur le support, c'est lui qui doit assurer toutes les protections résultant d'une réaction chimique avec la surface du support. En particulier, c'est lui qui assure la protection anti-corrosion d'un système, empêche les larves des insectes de se nicher dans le bois, et l'humidité de pénétrer dans le support. Enfin, il constitue une couche d'accrochage sur un support lisse pour la finition.

Le primer possède un P.V.C. très élevé, ce qui signifie un grand nombre de pigment. C'est grâce à ce P.V.C. élevé qu'il peut opacifier un fond. Il est d'aspect mat, et relativement poreux. Il n'est généralement pas teinté, ou alors il a une couleur neutre qui ne passe pas dans les couches supérieures du système.

ANNEXE A

3.1.3. Les surfacers.

Le surfacer, ou couche intermédiaire, a pour but de donner au système une épaisseur suffisante pour lui permettre de résister assez longtemps aux intempéries, à l'abrasion, et de protéger efficacement le support contre les infiltrations d'eau et de CO₂.

Généralement, il consiste en un primer d'accrochage étendu sur un primer de protection, ou en une couche supplémentaire du produit de finition.

Il n'existe pas de surfacers typiques; les surfacers ne sont pas des produits spéciaux.

3.1.4. Les finitions.

La finition est la ou les deux dernière(s) couche(s) du système. C'est elle qui est en contact avec l'environnement ambiant, et donne son aspect brillant, satiné ou mat au système.

Elle doit absolument répondre aux exigences de résistance et d'aspect du peintre.

Les exigences de protection seront prises en charge par le primer et/ou le surfacer.

3.1.5. Remarque.

Il existe des produits qui sont à la fois primers et finitions. La différenciation se fait au niveau du rôle qu'ils jouent dans le système.

3.2. CONSTRUCTION D'UN SYSTEME DE PEINTURE.

Un système de peinture vise à transformer l'état initial d'un support en un autre préalablement décrit. Ainsi, il faut que le système dans son ensemble satisfasse toutes les exigences du peintre. Ce système doit comprendre des traitements qui sont compatibles entre eux, c'est-à-dire qui ne se détruisent pas mutuellement, et qui sont compatibles avec l'état final désiré.

3.3. PROBLEMATIQUE DES COMPATIBILITES.

Toutes les couches d'un système doivent être compatibles entre elles, et la première couche doit être compatible avec le support. Il existe plusieurs types d'incompatibilités décrits par des règles précises.

3.3.1. Le PVC.

Le P.V.C. doit décroître en partant du primer vers la finition. Cette règle induit que l'on ne peut couvrir un produit brillant que d'un produit au moins aussi brillant que lui.

3.3.2. Le solvant.

Le solvant d'une couche ne peut altérer la couche de peinture qui précède. Dès lors, l'agressivité du solvant doit décroître en partant du primer vers la finition. Cas particulier de cette règle : le solvant du primer ne doit pas altérer le support.

3.3.3. La thermoplasticité.

La thermoplasticité doit croître ou rester constante en partant du primer vers la finition, ceci pour éviter des problèmes de craquelures lorsque le support bouge sous l'effet de la chaleur. Le petit schéma ci-dessous illustre cette règle.

3.3.4. Le liant.

Le liant doit, dans la mesure du possible, rester constant dans le système. De plus, le liant ne doit pas saponifier sur le pigment de la couche précédente. En particulier, un liant alkyde saponifie sur un pigment de zinc.

On peut formaliser les compatibilités du niveau du liant par la règle de la page suivante.

ANNEXE A

Soit E_i un état,
TRTi un traitement :

$$E_1 = E_F, E_n = E_I, 1 \leq i \leq n, E_i \xrightarrow{\text{TRT}_i} E_{i+1}$$

Règle générale : $\text{liant}(\text{TRT}_i) = \text{liant}(\text{TRT}_{i+1})$, $1 \leq i < n$

On peut aussi avoir :

- a) $\text{liant}(\text{TRT}_i) = \text{caoutchouc chloré}$
et $\text{liant}(\text{TRT}_j) = \text{copolymères vinyliques chlorés}$
avec $i = j+1$ ou $i = j-1$
- b) si $\text{liant}(\text{TRT}_i) = \text{résines alkydes}$
alors $\text{liant}(\text{TRT}_{i-1}) = \text{résines alkydes}$
ou caoutchouc chloré
ou huiles siccatives
ou résines oléostyrénées
- c) si $\text{liant}(\text{TRT}_i) = \text{résines époxydiques}$
alors $\text{liant}(\text{TRT}_{i-1}) = \text{résines époxydiques}$
ou " acryliques
ou " alkydes
ou caoutchouc chloré
ou copolymères vinyliques chlorés
ou PUR mono aliphatiques
ou " " aromatiques
ou " bicomposants
ou émulsions aqueuses
- d) si $\text{liant}(\text{TRT}_i) = \text{huiles siccatives}$
ou polymères azotés
ou polyvinyl butyral
ou résines alkydes
ou " oléostyrénées
ou " phénoliques
alors (($i < n-1$) ou (nature(support) <> galva
et <> zinc))
ou
(pigment(TRT_{i+1}) <> borate de zinc et
<> chromate de zinc et
<> phosphate de zinc et
<> oxyde de zinc et
<> zinc métal)

3.3.5. Le décapage.

Enfin, si aucun produit ne peut convenir, cas qui se produit lorsqu'on rénove un support déjà peint et que rien n'est compatible avec l'ancien produit, il faut décaper et reprendre à zéro.

ANNEXE A

3.4. EXEMPLE DE FICHE DE SYSTEME DE PEINTURE.

SYSTEME DE PEINTURE TRIMETAL POUR LE TRAITEMENT DES OBJETS METALLIQUES.

FER - ACIER - FONTE - ZINC - GALVANISE - ALUMINIUM - CUIVRE

Les fonds doivent être secs, dégraissés, dépoussiérés et débarrassés de la rouille non-adhérente.

PREPARATION DES FONDS

Fers neufs : dérouiller - décalaminer

Zinc, galvanisé, cuivre, laiton neufs : dégraisser au white spirit ou au trichloréthylène

Zinc, galvanisé anciens non-peints : laver à l'eau claire

Métaux déjà peints : retoucher les endroits dénudés avec la première couche

Peintures brillantes : dépolir au moyen de papier abrasif

TRAITEMENT DE SURFACE

EXTERIEUR

INTERIEUR

<u>TRAITEMENT DE SURFACE</u>		EXTERIEUR	INTERIEUR
Fers neufs	brillants	1c PLOMO	1c STELOXINE!
		2c PERMALINE	1c PERMALINE!
	½brillants	3c STELOXINE	2c STELOXINE!
Radiateurs	satinés	----	1c STELOXINE!
			1c TRISATIN !
Métaux déjà peints	brillants	1c PERMALINE	1c PERMALINE!
		1c STELOXINE	1c STELOXINE!
	½brillants	1c STELOXINE	1c STELOXINE!
	satinés	1c TRISATIN	1c TRISATIN !
Galvanisé, aluminium, zinc	brillants	1c MULTIPRIM	2c PERMACRYL!
		2c PERMACRYL	
Cuivre, aluminium	verniss	3c INCRALAC	2c INCRALAC !
Argenterie	verniss	----	1c INCRALAC !

ANNEXE A

RENDEMENTS

(exprimés en mètres carrés par litre de produit)

PLOMOFERRINE : 18

PERMALINE : 16

STELOXINE : 12

TRISATIN : 15

MULTIPRIMER : 10

PERMACRYL : 15

INCRALAC : 15 à 17

4. LE CONSEIL EN PEINTURE.

Le conseil en peinture est établi par un spécialiste sur demande d'un client. Celui-ci fixe ses exigences concernant l'état final de l'objet à peindre, selon un certain nombre de critères qui seront décrits plus bas. Le travail du spécialiste consiste à proposer un système de peinture qui respecte au mieux les exigences de son client. Si celles-ci ne peuvent être toutes satisfaites, l'expert devra essayer de trouver un compromis avec le client de façon à pouvoir satisfaire ses besoins.

4.1. CRITERES.

Les exigences de l'utilisateur sont décrites suivant deux catégories de critères : ceux qui affectent directement le support, et ceux qui affectent autre chose que le support.

Le premier groupe reprend les critères suivants :

- l'environnement : le support traité doit y résister.
- l'aspect : brillant ou non, aspect structuré ou non, opacité, ...
- les éléments contre lesquels il doit résister : les acides les infiltrations d'eau, ...
- les éléments contre lesquels il doit être protégé : la rouille, les insectes, ...
- la matière : bois et fer nécessiteront des traitements différents.
- les caractéristiques spéciales : antidérapant, ...

Chaque produit de la gamme rencontre au moins une de ces exigences.

Le second groupe reprend des critères du type prix de revient maximal, disponibilité immédiate du produit, durée maximale de la séquence des traitements, matériel disponible (tout le monde ne possède pas un pistolet), ...

4.2. COMMENT S'EN TIRER PRATIQUEMENT ?

4.2.1. Caractérisation de l'objet à peindre.

La première tâche est de caractériser l'état actuel de l'objet à peindre. Il faut savoir s'il est abîmé ou pas, réparable ou pas. Dans le cas où il est abîmé et réparable, l'expert devra donner les phases de préparation et/ou d'enduisage du support. Dans le cas où l'objet n'est pas abîmé, une préparation simple du genre dégraissage

ANNEXE A

suffira à le rendre recouvrable.

4.2.2. Caractérisation des exigences du client.

L'expert doit caractériser les exigences du client suivant la nomenclature décrite au point 4.1 .

A partir de là, il faut concilier état initial, état désiré, exigences du client et produits possibles.

4.2.3. Adoption d'un système.

L'expert doit maintenant tout "mettre en musique", compte tenu de ce dont il dispose et des exigences du client.

De toute façon, le système adopté doit répondre au nombre maximal de critères, le mieux étant qu'il réponde à tous. Si cette dernière situation ne peut être atteinte, l'expert doit rediscuter avec son client de ses exigences, et essayer de trouver un compromis avec lui de façon à lui conseiller un système. Le client pourrait être amené, par exemple, à adopter un système d'aspect satiné plutôt que mat.

multiprimer

couche primaire à multiples usages

MULTIPRIMER est une couche primaire qui allie rapidité de séchage et adhérence exceptionnelle sur les supports les plus divers. Il convient particulièrement comme couche primaire pour peintures en dispersion tel le PERMACRYL et à l'intérieur pour toute finition, tant à base de white spirit qu'à base d'eau.

**Séchage très rapide.
Intérieur et extérieur.
Diluable à l'eau.**



MULTIPRIMER est exempt d'huiles siccatives et n'a par conséquent **pas d'odeur**.



MULTIPRIMER **sèche en 30 minutes** et est ponçable après 2 heures.



MULTIPRIMER permet d'achever **un système complet de peinture, pour intérieur, en une journée** par application de 1 ou 2 couches de primer, suivies de la finition choisie.



MULTIPRIMER **s'applique très facilement**, même sur boiserries légèrement humides.



MULTIPRIMER **adhère parfaitement** sur supports divers tels que tuyaux en PVC, polyester, céramique, galvanisé, zinc, aluminium, verre,...



MULTIPRIMER **évite le saignement** des matières colorantes contenues dans des boiserries neuves ou dénudées.

multiprimer

fiche technique 1.1.25

Mode d'application

- Application : pistolet conventionnel, airless (pression 150 bars - tête 18/81).
- Dilution :
 - pistolet : prêt à l'emploi, ne jamais diluer
 - airless : 5 % d'eau.
- Température d'application : min. 10°C.
- Temps de séchage à 20°C et 60 % H.R. :
 - hors collant : 20 min.
 - recouvrable :
 - par 2^e couche de MULTIPRIMER : 2 h.
 - par une couche synthétique, diluée au white spirit : ± 3 h., uniquement pour l'intérieur
 - par une couche de PERMACRYL : ± 5 h.
- Rendement : 10 m²/L sur support lisse.
- Nettoyage du matériel : eau, directement après l'emploi.
- Précaution : craint le gel, ne pas appliquer en dessous de 10°C.

Préparation des fonds

Le support doit être propre et cohérent. En présence d'algues, traiter au PARAMOSS.

- Béton, enduit de ciment, maçonnerie, amiante ciment : dépoussiérer. Sur supports poudreux ou non cohérents : appliquer 1 couche de STELFON.
- Bois neuf, panneaux d'agglomérés, boiserie sensibles aux cryptogames : dépoussiérer.
- Bois exotiques résineux (p.ex. merbau, afzélia) : préparer le fond avant d'appliquer le MULTIPRIMER afin de neutraliser l'action de ces résines sur le système de peinture.
- Acier galvanisé neuf, zinc, métaux non ferreux, PVC rigide, ... : dégraisser au solvant.
- Galvanisé et zinc anciens, polyester, verre, céramique : laver à l'eau additionnée de détergent.

Composition du système

Après préparation adéquate des fonds appliquer 1 ou 2 couches de MULTIPRIMER suivies de 1 ou 2 couches de finition en tenant compte de l'intervalle de couche minimum recommandé ci-dessous.

Extérieur :

mat :	STELATEX :	env. 3 h.
satiné :	STELACRYL VILLA :	env. 4 h.
brillant :	PERMACRYL :	env. 5 h.

Intérieur :

mat :	STELATEX :	env. 2 h.
satiné :	TRISATIN :	env. 3 h.
brillant :	PERMALINE :	env. 4 h.
	PERMACRYL :	env. 5 h.

Caractéristiques

Classification :	Afnor NF T 30-003 : I - classe 7 b ₂ .
Liant :	polymère 100 % acrylique.
Pigments :	oxyde de titane et pigments lamellaires.
Solvant :	eau.
Extrait sec en poids :	55 ± 3.
Extrait sec en volume :	40 ± 3.
Viscosité Epprecht à 20°C :	15 ± 2 d Pa.s.
Densité :	1,35 ± 0,1.
Brillant Gardner 60° :	15 ± 5
85° :	45 ± 5.
Epaisseur du film sec :	40 µ.

trimetal paint



BELGIUM
FRANCE
NEDERLAND
ITALIA

Emmanuellaan 1 - 1830 MACHELEN
70, av. du Pont de St-Denis - 92.234 GENNEVILLIERS
Viaductweg 78 - 6222 AB MAASTRICHT
17047 VADO LIGURE

Nederlandse tekst
op aanvraag

plomoferrine n & s

primers antirouilles semi-synthétiques

Les formulations antirouilles à base d'oxyde de fer jouissent d'une longue expérimentation et permettent ainsi une protection durable à tous les métaux ferreux. PLOMOFERRINE «N» et «S», primers antirouilles à base de minium de plomb, appartiennent à cette catégorie de produits. L'expérience est leur garantie de qualité.

Pour métaux ferreux.
Au minium de plomb.
Haut pouvoir couvrant.

Type spécial «S»
Inhibiteur incorporé.
Pour fond oxydé.



PLOMOFERRINE convient pour le traitement de **tous ouvrages en acier** tels charpentes, ponts métalliques, réservoirs de stockage, châssis de fenêtres, portes, poutrelles, etc.



PLOMOFERRINE se caractérise par une application aisée et un pouvoir couvrant très élevé, d'où un **rendement très favorable**.



PLOMOFERRINE est doté d'un **excellent arrondi** permettant une protection antirouille uniforme de toute la surface à traiter.



PLOMOFERRINE «N» assure une bonne **résistance** aux conditions atmosphériques et tout spécialement **en ambiance humide et industrielle**.



PLOMOFERRINE «S», plus fluide et plus pénétrant que le type «N», est doté d'un **inhibiteur incorporé**.



PLOMOFERRINE «S» peut être appliqué **sur surfaces imparfaitement dérouillées**. Il est donc spécialement adapté pour le traitement des métaux ferreux que l'accès ou les conditions de travail rendent difficiles à dérouiller. Un traitement ultérieur au PLOMOFERRINE «N» renforce la protection générale.

plomoferrine n & s

fiche technique 1.2.02

Mode d'application

- Application :
 - « S » : uniquement à la brosse, ce procédé étant le seul qui permette de tirer le maximum de profit du pouvoir mouillant des inhibiteurs.
 - « N » : brosse, rouleau, pistolet conventionnel, airless.
- Dilution :
 - brosse, rouleau, airless (pression 150 bars, 18/80 buse) : prêt à l'emploi
 - pistolet : DILUANT SYNTHETIQUE TPC, « S » 10 % et « N » 10-15 %.
- Température d'application : min. 0 °C - max. 35 °C.
- Séchage à 20 °C et 60 % H.R. :
 - « S » 4 h. et « N » 3 h. - recouvrable : 15 h.
- Rendement : 15 à 18 m²/L.
- Coloration : rouge-brun.
- Nettoyage du matériel :
 - « S » : STELCLEANER ou white spirit - « N » : white spirit.

Préparation des fonds

- Les fonds doivent être secs, soigneusement dépoussiérés par brossage et dégraissés.
- La calamine et la rouille peu adhérente doivent obligatoirement être éliminées.
- Les métaux difficiles à dérouiller, en atmosphère rurale ou urbaine peu agressive, seront d'abord traités au PLOMOFERRINE « S », puis recouverts d'une couche de PLOMOFERRINE « N ».
- En atmosphère industrielle, un traitement des fonds St2 ou Sa 2^{1/2} est recommandé avant application du PLOMOFERRINE « N ».

Composition du système

PLOMOFERRINE « N » résiste plusieurs mois à l'extérieur non recouvert. Pour une bonne protection, il est toutefois nécessaire de le recouvrir de finitions d'excellente qualité.

Couches anti-rouilles

Atmosphère rurale ou urbaine peu agressive :

- acier neuf préparé :
 - 1 c. PLOMOFERRINE « N », 30 microns sec
- acier rouillé préparé :
 - 1 c. PLOMOFERRINE « S », 20 microns sec et
 - 1 c. PLOMOFERRINE « N », 30 microns sec

Atmosphère urbaine agressive ou industrielle :

- appliquer une couche supplémentaire de PLOMOFERRINE « N » afin d'augmenter le pouvoir garnissant du système.

Couche intermédiaire

1 c. PRIMER SURFACER, 30 microns sec.

Couche finition

1 à 2 c. PERMALINE ou similaire, 30 microns sec.

Caractéristiques

	type « S »	type « N »
Classification	Afnor NF T 30-003 : I - classe 4a / 2a	
Liants	résines glycérophthaliques et huiles siccatives	
Pigments	minium de plomb - oxyde de fer	
Solvants	hydrocarbures aliphatiques	
Extrait sec en poids	79 ± 3	83 ± 3
Extrait sec en volume	55 ± 2	58 ± 2
Viscosité Afnor IV - 20 °C	35 s.	75 s.
Densité	1,70 ± 0,10	2,04 ± 0,10
Brillant Gardner 60 °	50 ± 10	45 ± 10
Point éclair	supérieur à 35 °C	supérieur à 43 °C
Epaisseur du film humide	35 ± 5 μ	50 ± 5 μ
Epaisseur du film sec	20 ± 5 μ	30 ± 5 μ
Finesse Hegman	minimum 4,5	
Elasticité Erichsen après 7 jours à 20 °C	7	6,5
Dureté Persoz après 8 jours à 20 °C	100 s.	87 s.
Vieillessement accéléré	300 h inaltéré	
Température en service	maximum 65 °C	



trimetal paint

BELGIUM

Emmanuellaan 1 - 1830 MACHELEN

FRANCE

70, av. du Pont de Saint-Denis - 92234 GENNEVILLIERS

NEDERLAND

Viaductweg 78 - 6222 AB MAASTRICHT

ITALIA

17047 VADO LIGURE

Nederlandse tekst
op aanvraag

SUPPORTS	PREPARATION DE SURFACE	PRIMAIRE	INTERMEDIAIRE	FINITION AU CHOIX		REMARQUES
				SATINEE	BRILLANTE	
1) Métaux ferreux non-peints	Pr3 Pr2 Pr2	Steloxine Decor Plomoferrine S Plomoferrine S	Steloxine Decor Steloxine Decor Tixoplomo	Steloxine Decor Trisatin Auber 3	Permol Stellapaint	Intérieur : 1 couche de finition Extérieur : 2 couches de finition ou locaux humides.
	Pr3 Pr6	Rubbercoating LM Rubbercoating PZ	Rubbercoating TX	Rubbercoat.TX		Convient pour atmosphère urbaine marine et industrielle.
2) Métaux ferreux: entretien	Pr8	*	*	*	*	(*): tout système de la rubrique 1) préférence à Plomoferrine S et Tixoplomo.
3) Galvanisé neuf	Pr9 Pr10 Pr10	Multiprimer Rubbercoating PZ Wash Primer Stelnyl <small>Zincano</small>	Permacryl - Steloxine Decor	- Rubbercoating TX Steloxine Decor Trisatin Auber 3	Permacryl - Permol Stellapaint	
4) Galvanisé ancien, peu ou pas rouillé.	Pr11	Zincano	Steloxine Decor	Steloxine Decor	-	
5) Galvanisé ancien fortement rouillé.	Pr12	Plomoferrine S	Steloxine Decor	Steloxine Decor	-	
6) Aluminium, cuivre non-peints	Pr10 Pr10	Wash Primer Stelnyl Wash Primer Stelnyl	Stelprimer Steloxine Decor	Trisatin Auber 3 Steloxine Decor	Permol Stellapaint -	
7) Aluminium, cuivre peints	Pr13	Wash Primer Stelnyl (retouches)	*	*	*	(*): tout système de la rubrique 6
8) Aluminium, cuivre neufs à vernir	Pr10	-	-	-	Incralac	Int.: 1 couche/ Ext.: 2 couches. Sur alu: essai d'adhérence du vernis.
9) Cuivre oxydé, à vernir.	Pr14	-	-	-	Incralac	

ANNEXE 12

20

PREPARATION DE SURFACE
SYMBLE

DESCRIPTION

Pr0	Aucune préparation.
Pr1	Brossage léger. La brosse d'acier doit être passée avec une pression légère et un mouvement de va-et-vient sur la surface de l'acier, de sorte que chaque partie soit brossée deux fois. (norme suédoise St1).
Pr2	Raclage soigné au racloir en métal dur et brossage soigné à la brosse d'acier. La racloir élimine la pellicule de laminage, la rouille et les impuretés. Ensuite, la surface doit être brossée énergiquement, nettoyée avec un aspirateur, à l'air comprimé ou avec une brosse propre. Obtention d'un léger brillant métallique. (norme suédoise St2).
Pr3	Idem Pr2, mais le raclage et le brossage sont effectués dans un sens, puis une direction perpendiculaire à celui-ci. Obtention d'un brillant très net (norme suédoise St3).
Pr4	Sablage léger. Elimination de la calamine, rouille et impuretés. Teinte rousse (norme suédoise Sa1).
Pr5	Sablage soigné. Elimination de la calamine, rouille et impuretés. Surface grisâtre à nuances rousses (norme suédoise Sa2).
Pr6	Sablage très soigné. Surface grisâtre avec fines traces de rouille (norme suédoise Sa 2 1/2).
Pr7	Sablage poussé. Disparition totale de la rouille.
Pr8	Elimination des peintures non-adhérentes. Au minimum, préparation Pr2.
Pr9	Dégraissage au PIXOL. Lavage à l'eau.
Pr10	Dégraissage à l'essence légère.
Pr11	Dérouillage éventuel, dégraissage à l'essence légère, lavage à l'eau.
Pr12	Selon le degré d'enrouillement du support : sablage, grattage ou brûlage.
Pr13	Elimination et ponçage des peintures non-adhérentes. Dégraissage à l'essence légère. En atmosphère saline, lavage à l'eau.
Pr14	Nettoyage à l'aide d'un produit du commerce (Sidol ou similaire). Dégraissage à l'essence légère.

ANNEXE C

NORMES.

1. Norme AFNOR NFT 30 001 (Avril 1969) : définition des ambiances et atmosphères.

a. atmosphère (extérieur).

Par convention, on parlera d'atmosphère pour définir un milieu extérieur.

L'atmosphère est l'ensemble constitué par l'environnement gazeux, les systèmes d'aérosol et les variétés de poussières auxquels est exposé un feuil maintenu au contact du milieu ambiant. L'atmosphère est l'élément constitutif indispensable du micro-climat.

a) atmosphère rurale.

éloignée de toute activité chimique et industrielle, de tout centre urbain et non-conditionnée par le climat maritime (air relativement pur et dénué de poussières).

b) atmosphère industrielle.

caractérisée par la présence de poussières et fumées (produits de la combustion directe ou indirecte du carbone, du soufre, de l'azote, etc...) donnant aux condensats des réactions acides (pH<6).

c) atmosphère urbaine.

caractérisée, comme la précédente, par des poussières et des condensats de réaction acide (pH<6).

d) atmosphère marine.

prévaut sur les côtes et dans leur voisinage immédiat. Elle est caractérisée par la présence dans l'air d'aérosols d'eau de mer, par des brouillards plus ou moins salins et par l'absence de poussières. Les condensats sont en général basiques (pH>7).

ANNEXE C

b. ambiance(intérieur).

~~~~~

Par convention, on parlera d'ambiance pour définir un milieu intérieur.

a) non-agressive.

-----

ambiance qui ne comporte aucune vapeur acide ou basique, ni humidité permanente (exemple : atelier de stockage, atelier de mécanique).

b) moyennement agressive.

-----

ambiance qui comporte des vapeurs acides ou basiques à forte teneur, conjuguées en général avec des poussières corrosives.

c) très agressive.

-----

ambiance qui comporte des vapeurs acides ou basiques à haute teneur, conjuguées en général avec des poussières corrosives.

## ANNEXE C

### 2. Norme suédoise SIS 05.59.00 : définition des modes de préparation Sa3, Sa2½, Sa2, St3, St2.

La durabilité d'un feuil de peinture dépend dans une large mesure du traitement préliminaire du support à protéger. Le meilleur résultat possible sera atteint par un sablage approprié. Comme agents de sablage, le corindon, le quartz et la scorie de haut fourneau ont fait leurs preuves. Avec une grosseur de grain de 0,1 à 2 mm, on obtient une rugosité moyenne de 60 à 40 microns.

Conformément à l'échelle suédoise de mesure du degré d'enrouillement, on distingue quatre qualités de sablage.

Sa 1 : sablage léger : la calamine, la rouille et les corps étrangers écaillés sont enlevés; la surface présente encore une teinte rousse.

Sa 2 : sablage soigné : presque toute la calamine, la rouille et les corps étrangers sont enlevés; la surface présente un ton gris avec encore des nuances rousses.

Sa 2½ : sablage très soigné : la calamine et la peau de laminage ont disparus; les fines traces de rouille ancrées dans les anfractuosités des pores n'apportent que de légères nuances au ton de la surface qui doit être gris-uni.

Sa 3 : sablage extrêmement soigné : on obtient ainsi le métal pur; les derniers trous de calamine, de rouille et les corps étrangers ont disparu; la surface présente un ton uniformément métallique sans aucune nuance rousse.

Si la surface ne peut être sablée, on procède à un brossage.

St 2 : brossage soigné : raclage et brossage soigné de la surface à la brosse métallique, à la meule, etc; le but de ce traitement est d'éliminer les pellicules facilement détachable de calamine, de rouille et les corps étrangers. La surface doit présenter un léger brillant métallique.

St 3 : brossage très soigné : même chose que pour le précédent mais le brossage se fait dans deux sens perpendiculaires. La surface doit présenter un brillant métallique très net.

ANNEXE C

3. Norme DIN d'enrouillement :

| ° d'enrouillement | DIN 1   | DIN 2 | DIN 3 | DIN 4   | DIN 5 |
|-------------------|---------|-------|-------|---------|-------|
| % de rouille      | 0,5 - 1 | ± 5   | ± 15  | 30 - 40 | ± 50  |

4. Norme européenne :

| ° d'enrouillement | ° de protection | rouille en % |
|-------------------|-----------------|--------------|
| Re 0              | 10              | 0            |
| Re 1              | 9               | 0,05         |
| Re 2              | 8               | 0,5          |
| Re 3              | 7               | 1            |
| Re 4              | 6               | 3            |
| Re 5              | 5               | 8            |
| Re 6              | 4               | 15 - 20      |
| Re 7              | 3               | 40 - 50      |
| Re 8              | 2               | 75 - 85      |
| Re 9              | 1               | 95           |

ANNEXE D

```
/*          A. BASE DE FAITS          */
/*          A.1. POSTCONDITIONS      */

post( degreissage, [[nature, [acier, alu, galva, zinc]],
                   [carspec, [prepall],
                    [etatsurf, [corrosion_non, graisse_non,
                                poussiere_non, proprete_oui,
                                accrochage_oui, nettete_null]
                    ]]).

post( degreissage, [[nature, [metaux]],
                   [carspec, [prepall],
                    [etatsurf, [corrosion_non, graisse_non,
                                poussiere_non, proprete_oui,
                                accrochage_oui, nettete_bon_etat]]
                   ]):-
    etat_de_depart(Ed),
    valeur_item( [etatsurf, Es], Ed, 1),
    inclus( [nettete_bon_etat], Es) .

post( lavage_eau_claire, [[nature, [acier]],
                          [carspec, [prepall],
                           [etatsurf, [corrosion_non, graisse_oui,
                                         poussiere_non, proprete_oui,
                                         accrochage_oui, nettete_null]
                           ]
                          ]):-
    etat_de_depart( Ed),
    valeur_item( [etatsurf, Es], Ed, 1),
    not(appartient(corrosion_oui, Es)).

post( lavage_eau_claire, [[nature, [galva, zinc]],
                          [carspec, [prepall],
                           [etatsurf, [corrosion_non, graisse_oui,
                                         poussiere_non, proprete_oui,
                                         Accrochage_oui, nettete_null]
                           ]
                          ]):-
    etat_de_depart( Ed),
    valeur_item([etatsurf, Eds], Ed, 1),
    appartient(nettete_nu, Eds),
    etat_final( Ef),
    valeur_item([etatsurf, [recouvrable]], Ef, 1).
```



ANNEXE D

```
post( lavage_eau_claire, [[nature, [galva]],
                          [carspec, [prepall]],
                          [etatsurf, [corrosion_non, graisse_oui,
                                      poussiere_non, proprete_oui,
                                      Accrochage_A, nettete_nu]]
    ]):-
```

```
    etat_de_depart( Ed),
    valeur_item([etatsurf, Eds], Ed, 1),
    appartient(nettete_nu, Eds),
    etat_final( Ef),
    valeur_item([etatsurf, Es], Ef, 1),
    croiss(accro, A),
    appartient(Accrochage_A, A),
    appartient(Accrochage_A, Es).
```

```
post( lavage_eau_claire, [[nature, [zinc]],
                          [carspec, [prepall]],
                          [etatsurf, [corrosion_non, graisse_oui,
                                      poussiere_non, proprete_oui,
                                      accrochage_oui, nettete_nu]]
    ]):-
```

```
    etat_de_depart( Ed),
    valeur_item([etatsurf, Eds], Ed, 1),
    appartient(nettete_nu, Eds),
    etat_final( Ef),
    valeur_item([etatsurf, Es], Ef, 1),
    not(appartient( corrosion_oui, Es)).
```

```
post( lavage_eau_claire, [[nature, [metaux]],
                          [carspec, [prepall]],
                          [etatsurf, [corrosion_non, graisse_oui,
                                      poussiere_non, proprete_oui,
                                      Accrochage_A, nettete_bon_etat]]
    ]):-
```

```
    etat_de_depart( Ed),
    valeur_item( [etatsurf, Es], Ed, 1),
    inclus([nettete_bon_etat, proprete_non], Es),
    croiss(accro, A),
    appartient(Accrochage_A, A),
    appartient(Accrochage_A, Es) .
```

```
post( polissage, [[nature, [cuivre]],
                  [carspec, [prepall]],
                  [etatsurf, [corrosion_non, graisse_non,
                              poussiere_non, proprete_oui,
                              accrochage_oui, nettete_nu]]
    ]).
```

ANNEXE D

```

post( mordancage, [[nature,[galva, zinc]],
                  [carspec, [prepall]],
                  [etatsurf, [corrosion_non, graisse_oui,
                              poussiere_non, proprete_non,
                              accrochage_oui, nettete_nu]]
      ]):-
etat_de_depart( Ed),
valeur_item([etatsurf, Es], Ed, 1),
inclus( [accrochage_non, corrosion_non], Es).

post( lavage_eau_savon, [[nature, [zinc]],
                        [carspec, [prepall]],
                        [etatsurf, [corrosion_non, graisse_oui,
                                    poussiere_non, proprete_oui,
                                    accrochage_oui, nettete_nu]]
                        ]):-
etat_de_depart(Ed),
valeur_item( [etatsurf, Es], Ed, 1),
appartient( corrosion_oui, Es).

post( epoussetage, [[nature,[alu]],
                   [carspec, [prepall]],
                   [etatsurf, [corrosion_oui, graisse_oui,
                               poussiere_non, proprete_oui,
                               accrochage_douteux, nettete_nu]]
                   ]):-
etat_de_depart(Ed),
valeur_item([etatsurf, Es], Ed, 1),
inclus([corrosion_oui], Es).

post( epoussetage, [[nature,[alu]],
                   [carspec, [prepall]],
                   [etatsurf, [corrosion_non, graisse_oui,
                               poussiere_non, proprete_oui,
                               accrochage_douteux, nettete_nu]]
                   ]).

post( epoussetage, [[nature,[acier]],
                   [carspec, [prepall]],
                   [etatsurf, [corrosion_non, graisse_oui,
                               poussiere_non, proprete_oui,
                               accrochage_oui, nettete_nu]]
                   ]).

post( epoussetage, [[nature,[metaux]],
                   [carspec, [prepall]],
                   [etatsurf, [corrosion_non, graisse_oui,
                               poussiere_non, proprete_oui,
                               accrochage_oui, nettete_bon_etat]]
                   ]).

```

## ANNEXE D

```

post( epoussetage, [[nature,[galva, zinc]],
                  [carspec, [prepa]],
                  [etatsurf, [corrosion_non, graisse_oui,
                              poussiere_non, proprete_oui,
                              accrochage_oui, nettete_nu]]
      ]).

post( derouillage, [[nature,[acier, galva]],
                   [carspec, [prepa]],
                   [etatsurf, [corrosion_non, graisse_oui,
                               poussiere_oui, proprete_oui,
                               accrochage_oui, nettete_nu]]
      ]):-
    etat_de_depart(Ed),
    valeur_item([etatsurf, Es], Ed, 1),
    appartient(corrosion_oui, Es).

post( poncage, [[nature,[metaux]],
               [carspec, [prepa]],
               [etatsurf, [corrosion_non, graisse_oui,
                           poussiere_oui, proprete_oui,
                           accrochage_oui, nettete_bon_etat]]
      ]):-
    etat_de_depart(Ed),
    valeur_item([etatsurf, Es], Ed, 1),
    inclus([accrochage_non, nettete_bon_etat], Es).

post(auber3, [
    [nature, [metaux]],
    [env, [moyenagressif, urbain]],
    [aspect, [lisse, satine, filmogene, opaque]],
    [etatsurf, [recouvrable]],
    [resist, [lavage]]]).

post(aubexol, [
    [nature, [metaux]],
    [env, [tresagressif, indust]],
    [etatsurf, [recouvrable]],
    [aspect, [brillant, lisse, filmogene]],
    [carspec, [antiderapant]],
    [resist, [abrasionforte, solvants]],
    [protect, [corrosion]]]).

post(ergesol, [
    [nature, [metaux]],
    [env, [moyenagressif]],
    [etatsurf, [recouvrable]],
    [aspect, [satine, lisse, filmogene]],
    [carspec, [antiderapant]],
    [resist, [abrasionforte]]]).

```

ANNEXE D

```
post(incralac, [
    [nature, [cuivre,zinc,plomb,alul]],
    [env, [urbain,marin,moyenagressif]],
    [etatsurf, [recouvrable]],
    [aspect, [vernis,brillant,filmogene,lisse]],
    [resist, [abrasionmoyenne]],
    [protect, [corrosion]]]).
```

```
post(multiprimer, [
    [nature, [metaux]],
    [env, [urbain,moyenagressif]],
    [carspec, [primer]],
    [etatsurf, [recouvrable]],
    [aspect, [lisse, mat]]]).
```

```
post(permacryl, [
    [nature, [metaux]],
    [env, [urbain,moyenagressif]],
    [etatsurf, [recouvrable]],
    [aspect, [brillant,filmogene,lisse]],
    [resist, [eau]]]).
```

```
post(permaline, [
    [nature, [metaux]],
    [env,[urbain,moyenagressif]],
    [etatsurf, [recouvrable]],
    [aspect, [lisse,brillant,filmogene]],
    [resist, [lavage]]]).
```

```
post(plomoferrinen, [
    [nature, [acier,galva]],
    [env, [indust,moyenagressif]],
    [etatsurf, [recouvrable]],
    [aspect, [lisse,mat,filmogene]],
    [protect, [corrosion]]]).
```

```
post(plomoferrines, [
    [nature, [acier,galva]],
    [env, [indust,moyenagressif]],
    [etatsurf, [recouvrable]],
    [aspect, [lisse,mat,filmogene]]]).
```

```
post(primersurfacer, [
    [nature, [metaux]],
    [env, [moyenagressif,urbain]],
    [carspec, [primer]],
    [etatsurf, [recouvrable]],
    [aspect, [lisse,mat,filmogene]]]).
```

```
post(rubbercoating,[
    [nature, [galva, acier, zinc]],
    [env, [tresagressif, indust]],
    [carspec, [primer]],
    [etatsurf, [recouvrable]],
    [aspect, [mat,lisse,filmogene]],
```

## ANNEXE D

```

[protect, [corrosion]]]).

post(steldurplus, [
  [nature, [metaux]],
  [env, [tresagressif, indust]],
  [carspec, [antiderapant]],
  [etatsurf, [recouvrable]],
  [aspect, [satine, lisse, filmogene]],
  [resist, [abrasionforte, solvants]]]).

post(stellor, [
  [nature, [metaux]],
  [env, [tresagressif, urbain]],
  [etatsurf, [recouvrable]],
  [aspect, [brillant, lisse, filmogene]],
  [resist, [solvants, lavage]]]).

post(steloxine, [
  [nature, [acier, galva]],
  [env, [moyenagressif, urbain]],
  [etatsurf, [recouvrable]],
  [aspect, [satine, lisse, filmogene]],
  [protect, [corrosion]],
  [resist, [eau]]]).

post(trichrom, [
  [nature, [metaux]],
  [env, [tresagressif, marin, urbain]],
  [etatsurf, [recouvrable]],
  [aspect, [mat, lisse, filmogene]],
  [protect, [corrosion]]]).

post(trisatin, [
  [nature, [metaux]],
  [env, [urbain, moyenagressif]],
  [etatsurf, [recouvrable]],
  [aspect, [satine, lisse, filmogene]],
  [resist, [lavage, abrasionlegere]]]).

post(washprimer, [
  [nature, [acier, galva]],
  [env, [moyenagressif, urbain]],
  [etatsurf, [recouvrable]],
  [aspect, [satine, lisse, filmogene]],
  [protect, [corrosion]]]).

post(zincano, [
  [nature, [acier, galva]],
  [env, [moyenagressif, urbain]],
  [etatsurf, [recouvrable]],
  [aspect, [mat, lisse, filmogene]],
  [protect, [corrosion]]]).

```

ANNEXE D

```

post(zincage, [
    [nature, [acier,galva]],
    [env, [moyenagressif,urbain]],
    [etatsurf, [recouvrable]],
    [aspect, [mat,lisse,filmogene]],
    [protect, [corrosion]]]).

/*          A.2. PRECONDITIONS          */
pre( polissage, [
    [etatsurf, Es]
]):-
    etat_de_depart( Ed),
    extnat( Ed, [cuivre]),
    valeur_item([etatsurf, Es], Ed, 1).

pre( degraissage, [
    [etatsurf, [corrosion_oui,graisse_oui,
                poussiere_non,proprete_oui,
                accrochage_douteux,nettete_nu]]
]):-
    etat_de_depart(Ed),
    extnat( Ed, [alu]),
    valeur_item([etatsurf, Es], Ed, 1),
    appartient( corrosion_oui, Es).

pre( degraissage, [
    [etatsurf, [corrosion_non,graisse_oui,
                poussiere_non,proprete_oui,
                accrochage_douteux,nettete_nu]]
]):-
    etat_de_depart(Ed),
    extnat( Ed, [alu]).

pre( degraissage, [
    [etatsurf, [corrosion_non, graisse_oui,
                poussiere_non, proprete_oui,
                accrochage_oui, nettete_nu]]
]):-
    etat_de_depart( Ed),
    extnat( Ed, [acier]),
    valeur_item([etatsurf, Es], Ed, 1),
    appartient(nettete_nu, Es).

pre( degraissage, [
    [etatsurf, [corrosion_non, graisse_oui,
                poussiere_non, proprete_oui,
                accrochage_oui, nettete_bon_etat]]
]):-
    etat_de_depart( Ed),
    extnat( Ed, [Nature]),

```

ANNEXE D

```

appartient( Nature, [acier, alu, cuivre, galva, zinc])
valeur_item([etatsurf, Es], Ed, 1),
appartient(netttete_bon_etat, Es).

pre( degraissage, [
    [etatsurf, [corrosion_non, graisse_oui,
                poussiere_non, proprete_oui,
                accrochage_oui, netttete_nu]]
    ]):-
    etat_de_depart(Ed),
    extnat( Ed, [Nature]),
    appartient( Nature, [galva, zinc]),
    valeur_item([etatsurf, Es], Ed, 1),
    appartient(netttete_nu, Es).

pre( lavage_eau_claire, [
    [etatsurf, [corrosion_non, graisse_oui,
                Poussiere_A, proprete_non,
                Accrochage_B, netttete_bon_etat]]
    ]):-
    etat_de_depart( Ed),
    extnat( Ed, [Nature]),
    membre( Nature, [metaux]),
    valeur_item( [etatsurf, Es], Ed, 1),
    appartient(netttete_bon_etat, Es),
    croiss(accro, B),
    appartient(Accrochage_B, B),
    appartient(Accrochage_B, Es),
    croiss(pouss, A),
    appartient(Poussiere_A, A),
    appartient(Poussiere_A, Es).

pre( lavage_eau_claire, [
    [etatsurf, [corrosion_non, graisse_oui,
                poussiere_oui, proprete_non,
                Accrochage_A, netttete_bon_etat]]
    ]):-
    etat_de_depart( Ed),
    extnat( Ed, [Nature]),
    membre( Nature, [metaux]),
    valeur_item([etatsurf, Es], Ed, 1),
    appartient(netttete_bon_etat, Es),
    croiss(accro, A),
    appartient(Accrochage_A,A),
    appartient(Accrochage_A, Es).

```

ANNEXE D

```
pre( lavage_eau_claire, [
    [etatsurf, [corrosion_non, graisse_oui,
                Poussiere_A, proprete_non,
                accrochage_oui, nettete_null]
    ]):-
    etat_de_depart( Ed),
    extnat( Ed, [Nature]),
    appartient( Nature, [acier, galva, zinc]),
    appartient(accrochage_oui, Es),
    croiss(pouss, A),
    appartient(Poussiere_A, A),
    appartient(Poussiere_A, Es).
```

```
pre( lavage_eau_claire, [
    [etatsurf, [corrosion_non, graisse_oui,
                poussiere_non, proprete_non,
                accrochage_oui, nettete_null]
    ]):-
    etat_de_depart( Ed),
    extnat( Ed, [Nature]),
    appartient( Nature, [galva, zinc]),
    valeur_item( [etatsurf, Es], Ed, 1),
    inclus([accrochage_non, nettete_nu], Es).
```

```
pre( mordancage, [
    [etatsurf, [corrosion_non, Graisse_A,
                Poussiere_B, Proprete_C,
                accrochage_non, nettete_null]
    ]):-
    etat_de_depart( Ed),
    extnat( Ed, [Nature]),
    appartient( Nature, [galva, zinc]),
    valeur_item([etatsurf, Es], Ed, 1),
    inclus([nettete_nu, corrosion_non], Es),
    croiss(graissse, A),
    appartient(Graissse_A, A),
    appartient(Graissse_A, Es),
    croiss(pouss, B),
    appartient(Poussiere_B, B),
    appartient(Poussiere_B, Es),
    croiss(prop, C),
    appartient(Proprete_C, C),
    appartient(Proprete_C, Es).
```



ANNEXE D

```
pre( lavage_eau_savon, [
    [etatsurf, [corrosion_non, Graisse_A,
                Poussiere_B, Proprete_C,
                accrochage_oui, nettete_null]
    ]):-
    etat_de_depart( Ed),
    extnat( Ed, [zinc]),
    valeur_item([etatsurf, Es], Ed, 1),
    appartient(nettete_nu, Es),
    croiss(graiss, A),
    appartient(Graisse_A, A),
    appartient(Graisse_A, Es),
    croiss(pouss, B),
    appartient(Poussiere_B, B),
    appartient(Poussiere_B, Es),
    croiss(prop, C),
    appartient(Proprete_C, C),
    appartient(Proprete_C, Es).
```

```
pre( epoussetage, [
    [etatsurf, [corrosion_non, graisse_oui,
                poussiere_oui, proprete_oui,
                accrochage_oui, nettete_bon_etat]]
    ]:-
    etat_de_depart(Ed),
    extnat( Ed, [Nature]),
    membre( Nature, [metaux]),
    valeur_item([etatsurf, Es], Ed, 1),
    appartient(nettete_bon_etat, Es).
```

```
pre( epoussetage, [
    [etatsurf, [corrosion_non, graisse_oui,
                poussiere_oui, proprete_oui,
                accrochage_oui, nettete_null]
    ]):-
    etat_de_depart( Ed),
    extnat( Ed, [acier]).
```

```
pre( epoussetage, [
    [etatsurf, [corrosion_non, graisse_oui,
                poussiere_oui, proprete_oui,
                accrochage_non, nettete_null]
    ]:-
    etat_de_depart( Ed),
    extnat( Ed, [Nature]),
    appartient( Nature, [galva, zinc]),
    valeur_item([etat_surf, Es], Ed, 1),
    appartient(accrochage_non, Es).
```

```
pre( epoussetage, [
    [etatsurf, [corrosion_oui, graisse_oui,
                poussiere_oui, proprete_oui,
                accrochage_douteux, nettete_null]
    ]:-
```

ANNEXE D

```

etat_de_depart(Ed),
extrnat( Ed, [alu]),
valeur_item([etatsurf, Es], Ed, 1),
appartient(corrosion_oui, Es).

pre( epoussetage, [
    [etatsurf, [corrosion_non, graisse_oui,
                poussiere_oui, proprete_oui,
                accrochage_douteux, nettete_null]
    ]):-
    etat_de_depart(Ed),
    extrnat( Ed, [alu]).

pre( epoussetage, [
    [etatsurf, [corrosion_non, graisse_oui,
                poussiere_oui, proprete_oui,
                accrochage_oui, nettete_null]
    ]):-
    etat_de_depart(Ed),
    extrnat( Ed, [Nature]),
    appartient( Nature, [galva, zinc]).

pre( derouillage, [
    [etatsurf, [corrosion_oui, graisse_oui,
                Poussiere_B, Proprete_C,
                accrochage_oui, nettete_null]
    ]):-
    etat_de_depart(Ed),
    extrnat( Ed, [Nature]),
    appartient( Nature, [galva, acier]),
    valeur_item([etatsurf, [corrosion_oui, graisse_oui,
                            Poussiere_B, Proprete_C,
                            accrochage_oui, nettete_null],

pre( poncage, [
    [etatsurf, [corrosion_non, graisse_oui,
                poussiere_non, proprete_oui,
                accrochage_non, nettete_bon_etat]
    ]):-
    etat_de_depart( Ed),
    extrnat( Ed, [Nature]),
    membre( Nature, [metaux]),
    valeur_item([etatsurf, Es], Ed, 1),
    appartient(proprete_non, Es).

pre( poncage, [
    [etatsurf, [corrosion_non, graisse_oui,
                Poussiere_A, proprete_oui,
                accrochage_non, nettete_bon_etat]
    ]):-

```

ANNEXE D

```

etat_de_depart( Ed),
extnat( Ed, [Nature]),
membre( Nature, [metaux]),
valeur_item([etatsurf, Es], Ed, 1),
appartient(proprete_oui, Es),
croiss(pouss, A),
appartient(Poussiere_A, A),
appartient(Poussiere_A, Es).

pre(auber3, [
    [etatsurf, [recouvrable]],
    [dernttrt, [auber3]]]).

pre(aubexol, [
    [etatsurf, [recouvrable]],
    [dernttrt, [aubexol]]]).

pre(ergesol, [
    [etatsurf, [recouvrable]],
    [dernttrt, [ergesol]]]).

pre(incralac, [
    [etatsurf, [recouvrable]],
    [dernttrt, [incralac]]]).

pre(multiprimer, [
    [etatsurf, [recouvrable]],
    [dernttrt, [multiprimer]]]).

pre(permacryl, [
    [carspec, [primer]],
    [etatsurf, [recouvrable]],
    [dernttrt, [permacryl]]]).

pre(permaline, [
    [carspec, [primer]],
    [etatsurf, [recouvrable]],
    [dernttrt, [permaline]]]).

pre(plomoferrinen, [
    [etatsurf, [recouvrable]],
    [dernttrt, [plomoferrinen]]]).

pre(plomoferrines, [
    [etatsurf, [recouvrable]],
    [dernttrt, [plomoferrines]]]).

pre(primersurfacer, [
    [etatsurf, [recouvrable]],
    [dernttrt, [primersurfacer]]]).

pre(rubbercoating, [
    [etatsurf, [recouvrable]],
    [dernttrt, [rubbercoating]]]).

```

ANNEXE D

```
pre(steldurplus, [
    [etatsurf, [recouvrable]],
    [derntrt, [steldurplus]]]).
```

```
pre(stellor, [
    [etatsurf, [recouvrable]],
    [derntrt, [stellor]]]).
```

```
pre(steloxine, [
    [etatsurf, [recouvrable]],
    [derntrt, [steloxine]]]).
```

```
pre(trichrom, [
    [etatsurf, [recouvrable]],
    [derntrt, [trichrom]]]).
```

```
pre(trisatin, [
    [etatsurf, [recouvrable]],
    [derntrt, [trisatin]]]).
```

```
pre(washprimer, [
    [etatsurf, [recouvrable]],
    [derntrt, [washprimer]]]).
```

```
pre(zincano, [
    [etatsurf, [recouvrable]],
    [derntrt, [zincano]]]).
```

```
pre(zincanoe, [
    [etatsurf, [recouvrable]],
    [derntrt, [zincanoe]]]).
```

/\* A.3. COMPOSITIONS \*/

```
comp(auber3, [[s,hydrocarbali],[p, [dioxydetitane, silicesdediatomee]],
    [l,alkyde],[pvc,0.305]]).
```

```
comp(aubexol, [[s,hydrocarbarometesthers],[p,[]],[l,purmono],
    [pvc,0.169]]).
```

```
comp(ergesol, [[s,hydrocarbali],[p,dioxydetitane],[l,alkyde],
    [pvc,0.261]]).
```

```
comp(incralac, [[s,hydrocarbarometesthers],[p,[]],[l,acrylique],
    [pvc,0.000]]).
```

```
comp(multiprimer, [[s,eau],[p,[dioxydetitane]],[l,acrylique],
    [pvc,0.289]]).
```

```
comp(permacryl, [[s,eau],[p,[dioxydetitane]],[l,acrylique],
    [pvc,0.206]]).
```

```
comp(permaline, [[s,hydrocarbali],[p,[]],[l,alkyde],
    [pvc,0.153]]).
```

ANNEXE D

```

comp(plomoferrinen, [[s,hydrocarbali],[p,[miniumplomb,oxydefer]],
                    [l,alkyde],[pvc,1.000]]).
comp(plomoferrines, [[s,hydrocarbali],[p,[miniumplomb,oxydefer]],
                    [l,alkyde],[pvc,1.000]]).
comp(primersurfacer, [[s,hydrocarbali],[p,[dioxydetitane,silicesdediatom
                    [l,alkyde],[pvc,0.355]]]).
comp(rubbercoating, [[s,chlores],[p,[ ]],[l,purmono],[pvc,1.000]]).
comp(steldurplus, [[s,hydrocarbarometesters],[p,[ ]],[l,purmono],
                    [pvc,0.145]]).
comp(stellor, [[s,hydrocarbali],[p,[oxydetitane]],[l,alkyde],
               [pvc,0.174]]).
comp(steloxine, [[s,hydrocarbali],[p,[phosphatezinc]],[l,alkyde],
                 [pvc,0.401]]).
comp(trichrom, [[s,hydrocarbali],[p,[chromatezinc,dioxydetitane,oxydefer
               [l,alkyde],[pvc,0.518]]]).
comp(trisatin, [[s,hydrocarbali],[p,[oxydetiatane,carbonatecalcium]],
                [l,alkyde],[pvc,0.385]]).
comp(washprimer, [[s,hydrocarbarometalcools],[p,[chromatezinc]],
                  [l,epoxyetphenolique],[pvc,1.000]]).
comp(zincano, [[s,hydrocarbarom],[p,[zinc]],[l,vinylchlore],
               [pvc,0.715]]).
comp(zincanoe, [[s,hydrocarbarom],[p,[zinc]],[l,epoxypolyamide],
                [pvc,1.000]]).
/*          A.4. CARACTERISTIQUES DES VALEURS DU VECTEUR D'ETAT          */
/*          A.4.1. ORDRE DES VALEURS DE CRITERE POUR CHAQUE          */
/*                                NOM DE CRITERE                                */
/*
croiss( forme, [horizontal,oblique,vertical]).
croiss( forme, [plan,tubulaire,angulaire]).
croiss( forme, [surfvershaut,surfversbas]).

croiss( env, [rural,urbain,indust]).
croiss( env, [peuagressif,moyenagressif,tresagressif]).

--

croiss( carspec, [salissant,peusalissant]).
croiss( carspec, [toxique,nontoxique]).
croiss( carspec, [derapant,antiderapant]).

croiss( etatsurf, [peurouille,voilerouille,rouilleimport,oxyde]).

```

## ANNEXE D

```

croiss( pouss, [poussiere_oui, poussiere_non]).
croiss( prop, [proprete_oui, proprete_non]).
croiss( graisse, [graisse_oui, graisse_non]).
croiss( accro, [accrochage_oui, accrochage_douteux, accrochage_non]).

croiss( resist, [rural,urbain,indust]).
croiss( resist, [marin]).
croiss( resist, [peuagressif,moyenagressif,tresagressif]).
croiss( resist, [abrasionlegere, abrasionmoyenne, abrasionforte]).
croiss( resist, [eau, lavage]).
croiss( resist, [solvants]).

/*          A.4.2. CLASSES DE VALEURS DE CRITERE POUR CHAQUE          */
                NOM DE CRITERE                                     */

itemvalclasse( aspect, [translucide, incolore, lisse, non-filmogenel], pe
itemvalclasse( aspect, [mat, satine, brillant, special, peint, vernis], r
itemvalclasse( aspect, [opaque, colore, structure, filmogenel, supp).
itemvalclasse( resist, [vapeau], perm).

/*          A.5. CARACTERISTIQUES DES COMPOSANTS          */
                */

/*          A.5.1. ECHELLE D'AGRESSIVITE DES SOLVANTS          */
                */
echelagr([[eau,1], [hydrocarbali, 2], [alcools, 3], [hydrocarbarom, 4],
        [hydrocarbarometalcools, 4], [esters, 5], [hydrocarbarometest
        [cetones, 6], [chlores, 7]]).

/*          A.5.2. ECHELLE DE THERMOPLASTICITE DES LIANTS          */
                */
echeltherm([[eau, 1], [caoutchlore, 2], [acrylique, 3], [alkyde, 4],
        [vinylchlore, 4], [purmono, 5], [epoxy, 6], [epoxyetphenoliq
        [epoxypolyamide, 6], [purbi, 7]]).

/*          A.5.3. LIANTS SAPONIFIABLES          */
                */
saponifiable( [alkyde, epoxyetphenolique]).

/*          A.5.4. PIGMENTS COMPOSES DE ZINC          */
                */
compzinc( [zinc, chromatezinc, phosphatezinc]).

```

ANNEXE D

```

/*          B. BASE DE REGLES                                */
/*          B.1. PRIMITIVES                                  */
/*          B.1.1. MANIPULATION DE LISTES : APPARTIENT
                                                CONCAT
                                                DEDOUBLE
                                                DIFF
                                                INCLUS
                                                INFEGAL
                                                INFEGAL AUX                                */

```

```

appartient( T, [T:Q]).
appartient( L, [T:Q]) :- appartient( L, Q).

membre( E1, L ) :- appartient(E1, L).
membre( Nat, [metaux]) :- appartient(Nat, [acier, galva, cuivre, zinc,
                                           alu, plomb]).

equivalent( [recouvrable], L ) :-
    inclus( L, [corrosion_non, graisse_non, poussiere_non,
               proprete_oui, accrochage_oui, nettete_nu,
               nettete_bon_etat]).

inclus([],S).
inclus([F:Qf], S) :- appartient(F,S),
    inclus(Qf,S).

concat([], [], []).
concat([A:L1], [], [A:L2]) :- concat(L1, [], L2).
concat([A:L1], [A:L2], C) :- concat(L1, L2, C).

dedouble( [], []).
dedouble( La, [B:Lb]) :- appartient( B, Lb),!,
    dedouble( La, Lb).
dedouble([A:L1], [A:L2]) :- dedouble( La, Lb).

/* diff( A,B,C) => C= A\B au sens mathematique avec eventuellement des d

diff([], B, []).
diff([A:L1], B, Lc) :- appartient( A, B),!,
    diff( La, B, Lc).
diff([A:L1], B, [A:L2]) :- diff(L1, B, Lc).

infegal(A,A,L) :- appartient(A,L).
infegal(A,B,[A:L]) :- appartient(B,L).
infegal(A,B,[T:Q]) :- infegal(A,B,Q).

infegalaux(E1,[Comp:C],L) :- infegal(E1,Comp,L);
    infegalaux(E1,C,L).

```

ANNEXE D

```

/*          B.1.2. MANIPULATION DE LISTE_DE_CRITERES : VALEUR_ITEM
                                                    EXTVALCRIT */

valeur_item([Item, Valeur], [], Res) :- Res is 2, !.
valeur_item([Item, Valeur], [[Item, Valeur]!L], Res) :- Res is 1, !.
valeur_item([Item, Valeur], [[Item, Valeur]!L], Res) :- Res is 0, !.
valeur_item([Item, Valeur], [L!L!], Res) :-
    valeur_item([Item, Valeur], L!, Res).

/*          B.1.3. PRIMITIVES D'EXTRACTION : EXTNAT
                                                    EXTFORME
                                                    ...
                                                    */

extpvc(Trt, Pvc) :- comp(Trt, C),
    valeur_item([pvc, Pvc], C, 1).

extliant(Trt, Liant) :- comp(Trt, C),
    valeur_item([l, Liant], C, 1).

extsol(Trt, Sol) :- comp(Trt, C),
    valeur_item([s, Sol], C, 1).

extpig(Trt, Pig) :- comp(Trt, C),
    valeur_item([p, Pig], C, 1).

extvalcrit(Nomcrit, L, Valcrit) :- valeur_item([Nomcrit, Valcrit], L, 1)
    !.
extvalcrit(Nomcrit, L, []) :- valeur_item([Nomcrit, Valcrit], L, 2), !.

extnat(Etat, Nature) :- extvalcrit(nature, Etat, Nature).
extforme(Etat, Forme) :- extvalcrit(forme, Etat, Forme).
extenv(Etat, Env) :- extvalcrit(env, Etat, Env).
extcarspec(Etat, Carspec) :- extvalcrit(carspec, Etat, Carspec).
extetatsurf(Etat, Etatsurf) :- extvalcrit(etatsurf, Etat, Etatsurf).
extas(Etat, Aspect) :- extvalcrit(aspect, Etat, Aspect).
extres(Etat, Resist) :- extvalcrit(resist, Etat, Resist).
extprot(Etat, Prot) :- extvalcrit(protect, Etat, Prot).
extderntrt(Etat, Dt) :- extvalcrit(derntrt, Etat, Dt).

extitemsvect([[nature, Nat], [forme, Forme], [env, Env], [carspec, Car],
    [etatsurf, Es], [aspect, As], [resist, Res], [protect,
    [derntrt, Dt]], Nat, Forme, Env, Car, Es, As, Res, Pro,

```



## ANNEXE D

```

extitemsvect( V, Nat, Forme, Env, Car, Es, As, Res, Pro, Dt) :-
    extnat( V, Nat),
    extforme( V, Forme),
    extenv( V, Env),
    extcarspec( V, Car),
    extetatsurf( V, Es),
    extas( V, As),
    extres( V, Res),
    extprot( V, Pro),
    extderntnt( V, Dt).

extlistitem( Type, Item, [], []) :- !.
extlistitem( Type, Item, L, []) :- not(itemvalclasse(Item, S, Type)), !.
extlistitem( Type, Item, [I|Linit], R) :-
    itemvalclasse( Item, S, Type),
    extlistaux( [I|Linit], R, S), !.

extlistaux( Init, [], []).
extlistaux( [], [], S).
extlistaux( [I|Linit], [I|Lres], S) :-
    appartient( I, S),
    extlistaux( Linit, Lres, S).
extlistaux( [I|Linit], Lres, S) :-
    extlistaux( Linit, Lres, S).

/*          B.2. REGLES DE CONSTRUCTION DE LA SOLUTION          */
/*          B.2.1. CHOIX DU TRAITEMENT                          */

optimise(Ed) :- extetatsurf( Ed, Es),
    equivalent([recouvrable], Es),
    assert(special(pas_prepa)).

optimise(Ed).

enleve :- special(pas_prepa),
    retract(special(S)).
enleve.

convient(L1, Es, Ef) :- extitemsvect( Es, Nats, Formes, Envs,
                                     Cars, Ess, Ass, Ress, Pros, Dts)
    extitemsvect( Ef, Natf, Formef, Envf, Carf, Esf, Asf, Re
                Prof, Dtf),

    faitavancer( Cars, Carf, Ass, Asf, Ress, Resf, Pros, Pro
                f,

    convientnat(Natf, Nats),
    convientforme(Formef, Formes),
    convientenv(Envf, Envs),
    convientcarspec(Carf, Cars),
    convientas(Asf, Ass),
    convientes(Esf, Ess),
    convientres(Resf, Ress),
    convientdt(Dtf, [L1]), !.

```

## ANNEXE D

```

faitavancer( [prepa], __, __, __, __, __, __ ) :- not(special(pas_prepa)).
faitavancer( Cars, Carf, Ass, Asf, Ress, Resf, Pros, Prof) :-
    avancc( Cars, Carf);
    avanca( Ass, Asf);
    avancr( Ress, Resf);
    avancp( Pros, Prof).

avancc( Cars, [F!Carf]) :- inclus( [F!Carf], Cars).

avanca(Ass, Asf) :- extlistitem(remp, aspect, Asf, R),
    extlistitem(supp, aspect, Asf, S),
    concat( As, R, S),
    appartient( A, As),
    appartient( A, Ass).

avancr(Ress, Resf) :- croiss(resist, L),
    appartient(A, Resf),
    infegalaux( A, Ress, L).

avancp(Pros, Prof) :- appartient(A, Prof),
    appartient(A, Pros).

convientnat([], S).
convientnat(F, []).
convientnat([Natf!N], S) :- membre(Natf, S).

convientforme([], S).
convientforme(F, []).
convientforme(F, [S!Formes]) :- croiss( forme, L),
    not(infegalaux( S, F, L)),
    convientforme( F, Formes).

convientenv([], S).
convientenv(E, []).
convientenv([E!F], S) :- croiss( env, L),
    infegalaux( E, S, L).

convientcarspec( [], S).
convientcarspec( F, S) :- inclus( F, S).

convientres([], S).
convientres([R!F], S) :- croiss(resist, L),
    infegalaux(R, S, L),
    convientres( F, S).

convientes(F, S) :- inclus(F, S);
    .. equivalent(F, S).

convientas([], S).
convientas(F, S) :- extlistitem( perm, aspect, F, Perm),
    extlistitem( remp, aspect, F, Remp),
    inclus( Remp, S),
    inclus( Perm, S).

```

ANNEXE D

```
convientas(F, []) :- extlistitem( perm, aspect, F, Perm),
                    diff( F, Perm, []).
```

```
convientdt( [], S) :- !.
```

```
convientdt( F, []) :- !.
```

```
convientdt( [F;Lf], [S;Ls]) :- compatliant( F, S),
                                compatsolv( F, S),
                                compatpigli( F, S),
                                compatpvc( F, S), !.
```

```
convientdt( [F], [S]) :- comp(F, _),
                        not(comp(S, _)).
```

```
compatliant( F, S) :- extliant( F, Lf),
                    extliant( S, Ls),
                    echeltherm(A),
                    valeur_item( [Lf, Nf], A, 1),
                    valeur_item( [Ls, Ns], A, 1),
                    Nf =< Ns.
```

```
compatsolv( F, S) :- extsol( F, Lf),
                    extsol( S, Ls),
                    echelagr(A),
                    valeur_item( [Lf, Nf], A, 1),
                    valeur_item( [Ls, Ns], A, 1),
                    Nf =< Ns.
```

```
compatpvc(F, S) :- extpvc(F, Pf),
                 extpvc(S, Ps),
                 Pf =< Ps.
```

```
compatpigli(F, S) :- extpig( S, Lp),
                    extliant( F, Lf),
                    ok( Lp, Lf).
```

```
ok( L, Liant) :- saponifiable(A),
                 not(appartient( Liant, A)).
```

```
ok([], L).
```

```
ok( [T;Q], Liant) :- compzinc(A),
                    not(appartient(T, A)),
                    ok( Q, Liant).
```

## ANNEXE D

```

/*          B.2.2. CALCUL DU NOUVEAU VECTEUR D'ETAT          */

calculn(Ec,
        Ass, Ress, Pros,
        Carp, Esp, Asp, Resp, Prop,Dtp,
        Asf, Resf, Prof) :-

        Ec = [[nature, Natf], [forme, Formef], [env, Envf],
              [carspec, Carp], [etatsurf, Esp], [aspect, Asc],
              [resist, Resc], [protect, Procl], [derntrt, Dtp]]

        etat_final( Ef),
        extnat( Ef, Natf),
        extforme(Ef, Formef),
        extenv( Ef, Envf),

        extlistitem(supp, aspect, Asf, Supp),
        extlistitem(perm, aspect, Asf, Perm),
        calculaux(Perm, Supp, Ass, Asp, Asc),

        extlistitem(perm, resist, Resf, Permr),
        calculaux(Permr, Resf, Ress, Resp, Resc),

/*
*/
        calculres(Resc, Resf, Ress, Resp, Permr),

        extlistitem(perm, protect, Prof, Permp),
        calculaux(Permp, Prof, Pros, Prop, Proc), !.

calculaux(Perm, Supp, Moins, Pre, Res) :-
        diff(Supp, Moins, Diffp),
        concat(Tem, Diffp, Perm),
        concat(Resul, Tem, Pre),
        dedouble(Res, Resul).

calculres(Resc, F, S, P, Perm) :- calculresaux( N, F, S),
        concat(Tem, N, Perm),
        concat(Resu, Tem, P),
        dedouble(Resc, Resu).

calculresaux([],[], S).
calculresaux(N, [Rf:F], Rs) :- croiss(resist, L),
        appartient(Rf, L),
        appartient(R, Rs),
        appartient(R, L),
        calculresaux(N, F, Rs), !.
calculresaux([R:IN], [R:F], Rs) :-calculresaux(N, F, Rs), !.

```

## ANNEXE D

```
/*          B.3. CONSTRUCTION DE LA SOLUTION          */
```

```
resolution( [], Ed, Ef ) :-
    [etatsurf, E], [aspect, F], [resist, G], [protect, H],
    [derntrt, Y],
    [[nature, A], [forme, B], [env, C], [carspec, D],
     [etatsurf, I], [aspect, X], [resist, G], [protect, H],
     [derntrt, Z]] :-
    convientes(I, E),
    extlistitem(perm, aspect, X, Perm),
    diff(X, Perm, F), !.
```

```
resolution( [L1:L2], Ed, Ef ) :-
    post(L1, Es),

    convient(L1, Es, Ef),

    pre( L1, Ep),

    extcarspec( Ep, Carp),
    extetatsurf( Ep, Esp),
    extas( Ep, Asp),
    extres( Ep, Resp),
    extprot( Ep, Prop),
    extderntrt( Ep, Dtp),

    extas( Es, Ass),
    extres( Es, Ress),
    extprot( Es, Pros),

    extas( Ef, Asf),
    extres( Ef, Resf),
    extprot( Ef, Prof),

    calculn( Ec,
             Ass, Ress, Pros,
             Carp, Esp, Asp, Resp, Prop, Dtp,
             Asf, Resf, Prof),

    resolution( L2, Ed, Ec).
```

```
solution([L1:L2], Ed, Ef) :- etat_final(Ef),
                             etat_de_depart(Ed),
                             optimise( Ed),
                             resolution([L1:L2], Ed, Ef),
                             enleve.
```

```
solution([], Ed, Ef).
```

## ANNEXE D

```
conseil :- entete,
          entdonnees,
          repeat,
              solution(Sol, Ed, Ef),
              sortresult(Sol),
          fail,
          retract(etat_de_depart(Ed)),
          retract(etat_final(Ef)).

/*          C. INTERFACE          */
/*          C.1. OUTILS POUR L'INTERFACE          */
/* 1. Sortie d'une phrase à l'écran*/
sortphr([]) :- nl.
sortphr([T:Q]) :- write(T),
                  tab(1),
                  sortphr(Q).

/* 2. Inversion de l'ordre des éléments d'une liste*/
renvliste([],L,L).
renvliste([T:Q], L, R) :- renvliste(Q, [T:L], R).

/* 3. Sortie d'une liste à l'écran*/
sortliste([]).
sortliste([T:Q]) :- write(T),
                    nl,
                    sortliste(Q).

/* 4. Formation de l'état de surface à partir de ses valeurs */
accouple(Cor, Gr, Pous, Prop, Acc, Net, [C, G, Po, Pr, A, N]) :-
    couplec(Cor, C),
    coupleg(Gr, G),
    couplepo(Pous, Po),
    couplepr(Prop, Pr),
    couplea(Acc, A),
    couplen(Net, N).

/* 5. Couplage d'un critère d'état de surface et de sa valeur */
couplec(oui, corrosion_oui).
```

## ANNEXE D

```

couplec(non, corrosion_non).

coupleg(oui, graisse_oui).
coupleg(non, graisse_non).

couplepo(oui, poussiere_oui).
couplepo(non, poussiere_non).

couplepr(oui, proprete_oui).
couplepr(non, proprete_non).

couplea(oui, accrochage_oui).
couplea(douteux, accrochage_douteux).
couplea(non, accrochage_non).

couplien(nu, nettete_nu).
couplien(bon_etat, nettete_bon_etat).

```

```

/*                C.2. INTERFACE                */

```

```

/* 1. Petite entête*/

```

```

entete :- sortphr(['Bonjour.', 'Veuillez', entrer, successivement]),
          sortphr(['l''etat', initial, et, 'l''etat', desire]),
          sortphr([du, support, apres, 'traitement.']).

```

```

/* 2. Entrée des données*/

```

```

entetatinit(Nat, Forme, Env) :-
    sortphr(['Veuillez', decrire, 'l''etat', initial, '.']),
    sortphr(['Pour', les, valeurs, 'possibles,',
            'reportez-vous', au, mode, 'd''emploi', joint]),
    sortphr([au, programme]),
    nl,
    write('Nature : '), read(Nat), nl,
    write('Forme : '), read(Forme), nl,
    write('Environnement : '), read(Env), nl,
    write('Etat'), tab(1), write(de), tab(1), write('surface
nl,
tab(5), write('corrosion : '), read(Cor), nl,
tab(5), write('graisse : '), read(Gr), nl,
tab(5), write('poussiere : '), read(Pous), nl,
tab(5), write('proprete : '), read(Prop), nl,
tab(5), write('accrochage : '), read(Acc), nl,
tab(5), write('nettete : '), read(Net), nl,nl,
write('Protections : '), read(Prot), nl,
accouple(Cor,Gr,Pous,Prop,Acc,Net,Etsurf),
assert(etat_de_depart( [[nature, Nat],
                        [forme, Forme],
                        [env, Env],
                        [carspec, []],

```

ANNEXE D

```

[etatsurf, Etsurf],
[aspect, []],
[resist, []],
[protect, Prot],
[derntrt, []]
1)).

entetatfin(Nat, Forme, Env) :-
    sortphr(['Veuillez', decrine, 'l''etat', 'final.']),
    sortphr(['Pour', les, valeurs, 'possibles,',
            'reportez-vous', au, mode, 'd''emploi', joint])
    sortphr([au, programme]),
    nl,
    write('Aspect : '), read(Asp), nl,
    write('Caracteristiques'), tab(1), write('speciales : ')
    read(Carspec), nl,
    write('Resistances : '), read(Resist), nl,
    write('Protections : '), read(Protfin), nl,
    assert(etat_final( [[nature, Nat],
                        [forme, Forme],
                        [env, Env],
                        [carspec, Carspec],
                        [etatsurf, [recouvrable]],
                        [aspect, Asp],
                        [resist, Resist],
                        [protect, Protfin],
                        [derntrt, []]
                        ])).

entdonnees :- entetatinit(Nat, Forme, Env),
              entetatfin(Nat, Forme, Env).

/* 3. Sortie des résultats*/

sortresult([]) :- sortphr(['Il', 'n''existe', pas, de, solution, a, votr
                        'probleme.']).

sortresult(Result) :- renvliste( Result, [], Res),
    sortphr(['La', suite, de, traitements, a, applique
            'est : ']),
    nl,
    sortliste(Res).

```