



HAL
open science

Contribution à la formulation des problèmes en conception de systèmes techniques. Etude basée sur la TRIZ

Sébastien Dubois

► **To cite this version:**

Sébastien Dubois. Contribution à la formulation des problèmes en conception de systèmes techniques. Etude basée sur la TRIZ. Sciences de l'ingénieur [physics]. INSA de Strasbourg, 2004. Français. NNT: . tel-00281657

HAL Id: tel-00281657

<https://theses.hal.science/tel-00281657>

Submitted on 23 May 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre :

École Doctorale Sciences Pour l'Ingénieur
ULP – INSA – ENGEES - URS

THÈSE

présentée pour obtenir le grade de

Docteur de l'Université Louis Pasteur – Strasbourg I
Discipline : Sciences pour l'Ingénieur
(spécialité Productique)

par

Sébastien DUBOIS

Contribution à la Formulation des Problèmes en Conception de
Systèmes Techniques
Etude basée sur la TRIZ

Soutenue publiquement le xx mois 2004

Membres du jury

Directeur de thèse : M. Philippe LUTZ, Professeur, Université de Franche Comté, Besançon

Directeur de thèse : M. François ROUSSELOT, Maître de Conférences, DE, Université Marc Bloch, Strasbourg

Rapporteur interne : M. Dominique VIAUD, Professeur, Université Louis Pasteur, Strasbourg

Rapporteur externe : M. Michel ALDANONDO, Professeur, Ecole des Mines, Albi

Rapporteur externe : M. Michel TOLLENAERE, Professeur, Ecole Nationale Supérieure de Génie Industriel, Grenoble

Examineur : M^{me} Rose DIENG, Directeur de Recherche INRIA, Unité de Sophia-Antipolis

REMERCIEMENTS

TABLE DES MATIERES

Remerciements.....	2
Table des matières	3
Introduction générale	6
I. La conception des systèmes techniques, un domaine de résolution de problèmes	11
I.1. La conception d'objets techniques, visions et consensus.....	14
I.1.1. Définition de la conception	14
I.1.2. La conception en tant que processus	16
I.1.3. Différentes vues de la conception.....	18
I.1.4. Des méthodes de conception variées.....	20
I.1.5. Positionnement vis-à-vis de la conception	23
I.2. Les méthodes traditionnelles de la conception.....	25
I.2.1. L'approche systématique de Pahl et Beitz.....	25
I.2.2. L'analyse de la Valeur, une approche centrée sur les fonctions	31
I.2.3. L'approche axiomatique de la conception.....	37
I.2.4. Le Quality Function Deployment.....	42
I.2.5. La TRIZ, théorie de résolution des problèmes inventifs	45
I.3. Définition de la problématique	49
I.3.1. Une vision multi-modulaire de la conception	49
I.3.2. Comparaison modulaire des approches	54
I.4. Positionnement de la problématique	57
II. La TRIZ une théorie centrée sur la résolution de problèmes.....	58
II.1. Le processus de résolution de problèmes de la TRIZ, une construction conjointe du problème et de sa solution	59
II.1.1. Le schéma général de la résolution de problèmes de la TRIZ	59
II.1.2. Le processus convergent	62
II.1.3. Un processus de construction progressif de la solution	65

II.2. Des concepts de base.....	66
II.2.1. Le Résultat Idéal Final	66
II.2.2. Les lois d'évolutions des systèmes techniques ([Altshuller, 1988; Salamatov, 1996])	68
II.2.3. Tout problème est une contradiction.....	74
II.2.4. Les ressources	77
II.2.5. L'inertie psychologique	79
II.3. Des méthodes de formulation et de positionnement de problèmes.....	81
II.3.1. L'approche multi-écrans, structure de l'esprit fort.....	81
II.3.2. Les standards de résolution	84
II.3.3. ARIZ	89
II.4. Des outils informatiques associés	95
II.4.1. TechOptimizer ([Invention Machine, 1999]).....	95
II.4.2. Innovation Workbench ([Ideation, 2000b])	97
II.4.3. Improver ([Ideation, 2000a]).....	99
II.4.4. Conclusion sur les outils informatiques	101
II.5. Conclusion.....	102
III. Constitution d'un modèle de formulation des problèmes par les outils de l'Intelligence Artificielle	106
III.1. Apports de l'Intelligence Artificielle pour la résolution de problèmes de conception	106
III.1.1. Les approches de la résolution de problème.....	106
III.1.2. La représentation des connaissances pour la conception inventive.....	116
III.1.3. Démarche de modélisation.....	121
III.2. un modèle de formulation des problèmes selon les cadres de la TRIZ	124
III.2.1. Identification des concepts.....	126
III.2.2. Construction du modèle de représentation des problèmes.....	134
III.2.3. Conclusions sur le modèle	146

IV. Exploitation et validation de l'outil	149
IV.1. Méthode de validation	149
IV.1.1. Objectifs de la validation	149
IV.1.2. Expérimentations de validation	152
IV.2. Illustration des expérimentations.....	153
IV.2.1. Validation de la convergence de la formulation.....	153
IV.2.2. Validation de l'apport du logiciel à la formulation de contradiction	156
IV.3. Limites	159
IV.3.1. Limites de validation	159
IV.3.2. Limites de l'interface.....	160
IV.3.3. Limites du modèle vis-à-vis de la TRIZ.....	162
IV.4. Apports et perspectives.....	163
IV.4.1. Apport technique : assistance à la formulation de problèmes	163
IV.4.2. Apport scientifique : formalisation de la TRIZ	164
IV.4.3. Perspectives de développement	165
IV.5. Conclusion	170
Conclusion générale.....	171
Bibliographie	174
Annexes	192

INTRODUCTION GENERALE

L'innovation est devenue un mot clé dans le vocabulaire industriel, car il est fortement rattaché à l'identité et au dynamisme des entreprises. Dans notre société de consommation, l'innovation est présente de façon constante et doit s'afficher. Si celle-ci peut être considérée comme intrinsèquement liée à l'homme de tout temps, les contraintes du monde actuel lui font revêtir une perception particulière, guidée par la notion de rentabilité. Sortir de nouveaux produits, se différencier par rapport aux autres, minimiser les coûts sont autant de contraintes qui imposent à l'entreprise d'accroître sa réactivité. L'innovation est caractérisée, dans [Roucoules et al., 2003], par trois axes : la nature de l'innovation (méthode, produit, service, ...), le succès rencontré par l'innovation (au sein d'une entreprise, sur un secteur d'activité, sur l'ensemble du marché) et, enfin, le degré de nouveauté.

Du point de vue de la recherche, traiter du problème de l'innovation et de sa gestion conduit à des approches fortement pluridisciplinaires : la conception, la gestion, les ressources humaines.

Nous situons notre action sur la conception inventive, qui s'intéresse à un mode d'innovation basé sur l'invention. Comment gérer l'activité de conception pour déclencher systématiquement l'inventivité ? La formalisation de démarches de conception inventive est un thème de recherche dont l'intérêt, autant industriel que scientifique, n'est pas remis en cause. Les travaux présentés dans cette thèse s'inscrivent dans cet esprit.

La problématique abordée

L'activité de conception représente un enjeu majeur pour les entreprises, dont la survie dépend bien souvent de la capacité à proposer des produits nouveaux ou développés à moindre coût. Dans ce contexte, la TRIZ propose une démarche intéressante par le fait qu'elle permet de cibler, pour un objectif donné, les problèmes à résoudre. Cette approche conduit à la formalisation d'une solution satisfaisant le problème identifié. De fait, elle se distingue des approches traditionnelles de la conception par une durée réduite de génération des concepts de solution et par un nombre restreint de solutions générées.

Si l'intérêt de l'application est aujourd'hui reconnu, son intégration pose problème. La difficile appropriation des méthodes de la TRIZ et leur délicate mise en œuvre peut s'expliquer de plusieurs faits :

- le manque d'ouvrages de référence en français, et leur nombre limité en anglais ;
- le faible nombre d'experts formés ;
- le manque de formalisation des méthodes de la TRIZ ;
- la complexité du lien entre les différentes méthodes.

Ces défaillances entraînent, par ailleurs, un manque de reconnaissance de la TRIZ au niveau scientifique, il est donc nécessaire d'entreprendre un travail de formalisation des méthodes. Le travail présenté ici a pour but de palier cette absence de reconnaissance, par la formalisation des concepts participant à la formulation des problèmes.

Le choix de centrer nos recherches sur la méthode de formulation des problèmes de la TRIZ résulte de la constatation que celle-ci est sans aucun doute l'une des plus délicate à mettre en œuvre, et pour laquelle une forte demande d'outils applicatifs émane des industriels. De plus, elle constitue la première phase de l'application de la TRIZ à un projet de conception, il est donc cohérent de démarrer par cette méthode.

L'approche proposée

Différents outils logiciels de mise en œuvre de méthodes de la TRIZ ou d'opérationnalisation d'outils existent sur le marché. Leur utilisation implique la nécessaire connaissance de la théorie, pour les exploiter avec efficacité. Ces logiciels ne cherchent pas à répondre au manque de formalisation que nous avons identifié comme clé de la difficulté d'appropriation de la TRIZ.

Pour parvenir à notre but, nous avons basé notre approche sur les travaux de formalisation des connaissances de résolution de problèmes issus de l'Intelligence Artificielle. La démarche suivie est passée par les étapes suivantes :

- identification des concepts propres à la méthode de formulation des problèmes, par l'étude des textes de référence définissant cette méthode ;
- modélisation des concepts en représentation centrée-objet, afin de construire une hiérarchie des concepts et d'explicitier les liens entre les différents concepts ;
- représentation du modèle centré-objet en logique de description pour la validation de la cohérence du modèle et de la cohérence de la hiérarchie du modèle ;

- constitution d'une interface d'exploitation du modèle centré-objet, permettant de guider la formulation du problème.

Les travaux réalisés

Le but de cette thèse est de répondre aux limites actuelles de la TRIZ en termes de formalisation. Devant l'étendue des notions de la TRIZ, nous concentrons notre travail sur les méthodes de formulation de problèmes. L'intérêt scientifique de notre démarche réside dans la méthodologie de formalisation que nous entreprenons, qui permet d'asseoir, sans ambiguïté, les connaissances extraites et de valider les bases de la TRIZ, dont le T signifie théorie, bien qu'elle ne soit pas reconnue comme telle, de par son manque de formalisme.

Soucieux de répondre également à un besoin industriel, nous développons nos travaux de manière à rendre le résultat de notre recherche opérationnel.

Dans ce double objectif, la mise en œuvre de la démarche présentée précédemment nous a permis d'aboutir aux résultats suivants :

- la constitution d'une ontologie¹ de la formulation des problèmes, selon les cadres de la TRIZ, qui permet de répondre à la problématique scientifique ;
- l'opérationnalisation de cette ontologie par la constitution d'un logiciel d'assistance à la formulation des problèmes, selon les cadres de la TRIZ.

Ces deux apports sont complémentaires, la conception du logiciel nécessitant de définir une stratégie d'instanciation. Cette stratégie définit la méthode utilisée pour formuler les problèmes.

¹ « Les ontologies sont un outil conceptuel de modélisation des connaissances, issu de l'Intelligence Artificielle. Elles fournissent une base cohérente pour construire, et une référence partagée sur laquelle s'aligner, sous la forme d'un vocabulaire conceptuel et consensuel, sur laquelle peuvent être construites des descriptions et peuvent se faire des communications. » [Gandon, 2002] Gandon, F. (2002) *Ontology Engineering: a Survey and a Return on Experience*, INstitut de Recherche en Informatique et Automatique (INRIA Sophia Antipolis), mars 2002.

La structuration du document

Cette thèse est constituée de quatre chapitres.

Le premier chapitre, « La conception des systèmes techniques, un domaine de résolution de problèmes », permet de positionner nos travaux dans le champ de la conception. Après avoir défini notre vision de la conception, nous présentons quelques-unes des approches méthodologiques de la conception. La troisième partie de ce chapitre montre que les différentes approches peuvent être complémentaires ; pour cela, nous proposons une description de la conception par modules. Cette description permet de mettre en évidence les apports des approches présentées. La comparaison s'appuie également sur le degré de formalisation des approches pour chaque module décrit. Nous concluons ce chapitre en présentant les apports potentiels de la TRIZ, qui présentent un intérêt pour la résolution des problèmes en conception, et mettons en lumière son manque de formalisation.

Le second chapitre, « La TRIZ une théorie centrée sur la résolution de problèmes », est une présentation détaillée de cette théorie. La première partie de ce chapitre montre l'intérêt de la démarche qu'elle préconise, reposant sur la construction d'un processus de résolution de problèmes qui converge vers la génération d'une solution. La seconde partie est dédiée à la présentation des concepts de base de la TRIZ qui permettent de guider le processus convergent. Nous décrivons ensuite les méthodes de formulation et de positionnement des problèmes, avant de présenter brièvement quelques outils informatiques disponibles sur le marché, issus de la TRIZ. La conclusion de ce chapitre a pour but de positionner, à nouveau dans une vision modulaire, les apports des logiciels vis-à-vis de la TRIZ et de mettre en exergue leurs limites.

La conclusion des deux premiers chapitres nous permet de définir proprement notre problématique, à la fois en termes scientifique et industriel.

L'objet du troisième chapitre, « Constitution d'un modèle de formulation des problèmes par les outils de l'Intelligence Artificielle », est de présenter la démarche que nous avons suivie et les travaux qui en résultent. Ce chapitre est décomposé en deux parties principales. La première a pour but de faire un bref état de l'art des approches d'Intelligence Artificielle pour la résolution de problème. Sur la base de cet état de l'art, nous explicitons la démarche de modélisation que nous avons suivie. La seconde partie du chapitre détaille la réalisation de cette démarche, qui nous a vu recourir à plusieurs outils d'Intelligence Artificielle. Cette seconde partie est décomposée selon les étapes suivantes :

- l'identification des concepts, retraçant l'analyse des textes entreprise et le choix des concepts qui seront inclus dans notre modèle ;
- la construction et la validation du modèle ;
- les conclusions sur le modèle afin d'explicitier, à la fois, les apports que constitue le modèle par rapport à l'existant théorique, et les difficultés que nous avons rencontrées dans la construction du modèle.

Enfin, le quatrième chapitre, « Exploitation et validation de l'outil », présente les expérimentations que nous avons pu mener avec le prototype de logiciel existant. Nous en présentons ensuite les limites et les apports forts. Nous finissons sur les perspectives de recherche qui nous semblent les plus importantes.

I. LA CONCEPTION DES SYSTEMES TECHNIQUES, UN DOMAINE DE RESOLUTION DE PROBLEMES

De tout temps, l'homme s'est intéressé à augmenter l'efficacité de son interaction avec l'environnement. Cherchant initialement à améliorer l'efficacité de ses propres fonctionnalités par la mise au point d'outils prolongeant les bras (voir les écrits de l'ethnologue Leroy-Gourhan, [Leroi-Gourhan, 1971]) (outils à trancher, à tisser, ...), portant la voix (cornes), la vue (signaux de fumée), ... Il s'est ensuite intéressé à la mise au point d'objets rendant des fonctionnalités nouvelles. Ces fonctionnalités ont été imaginées dans le but d'accroître le contrôle de l'homme sur son environnement : le silex pour allumer le feu, les lances pour se défendre et se nourrir, En fait l'homme, à travers ses inventions, cherche à satisfaire ses besoins primaires, de manière toujours plus forte. Maslow, psychologue américain, ([Maslow, 1954]) a classifié ces besoins hiérarchiquement (voir figure I.1.).

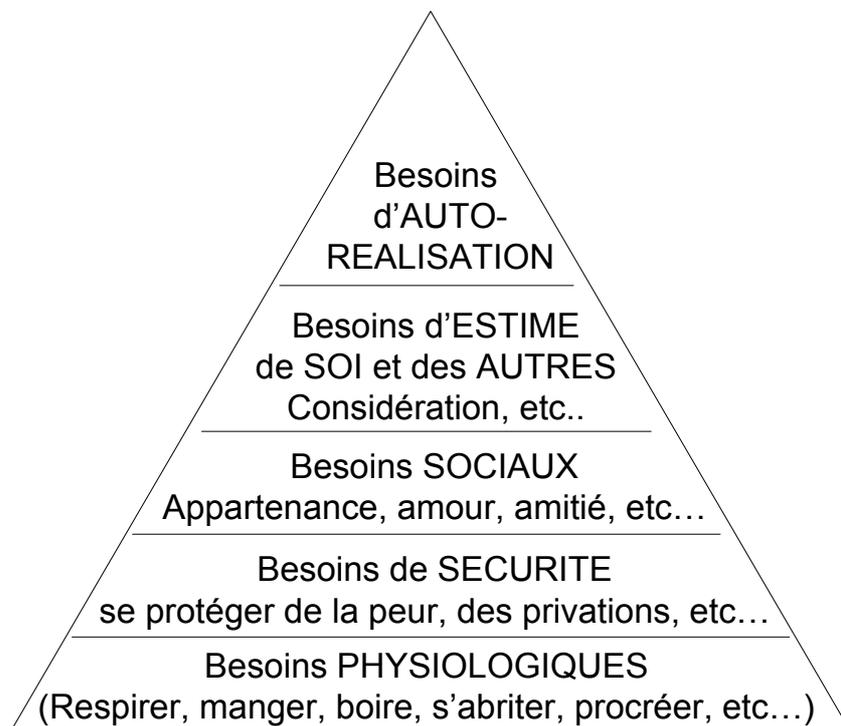


Figure I.1. La pyramide de Maslow, selon [Maslow, 1954]

L'homme préhistorique était donc déjà un créateur [Gogu, 2001], la création est une activité qui a toujours été propre à l'humanité. La conception est un processus permettant le passage de l'expression de besoins, qui sont traduits sous la forme de fonctions à réaliser, et pour lesquelles sont mis au point des systèmes répondant à leur satisfaction. La conception fut d'abord le résultat d'expériences menées par un « *homme de l'art* », comme le définit [Yannou, 2001],

personnage singulier pour qui la création est un acte inné non formalisable. Perrin (dans [Perrin, 2001]) nous montre l'évolution de la formalisation de la conception à partir des années 1940 et 1950, années d'émergence des premières méthodes de conception. D'acte personnel, d'apparition spontanée, la création devient méthodologique. Dans un premier temps, c'est le besoin de concevoir, tout en palliant les manques de ressources, à la fin de la seconde Guerre Mondiale, qui fait apparaître la nécessité de la formalisation.

Cet intérêt tardif pour la construction de méthodologies de conception explique bien le faible degré de formalisation de celles-ci. Gogu précise ainsi :

« Malgré l'existence multimillénaire de cette activité et malgré les avancements spectaculaires de la recherche scientifique dans la connaissance de l'homme, de son environnement et de ses outils, il y a un retard considérable dans la formalisation scientifique de la démarche méthodologique de l'innovation des systèmes techniques. » [Gogu, 2001]

La conception n'a pas été, pendant très longtemps, un objet de recherche, comme le rappelle Garro (dans [Garro, 2000]) ; le besoin d'explicitier, de transmettre les connaissances, n'était pas ressenti par des concepteurs désireux d'être les seuls garants de leur processus créatif.

Aujourd'hui la conception se décline en méthodes de conception, voire en théories de la conception ; toutefois il n'y a pas encore de consensus établi, du fait que le paradigme de la conception n'a pas encore été identifié. Cross, dans [Cross, 1992], décrit ainsi :

« Il manque un paradigme de la pensée de conception valable et simple. Les paradigmes simplificateurs qui ont été considérés par le passé - comme de voir la conception comme une simple résolution de problème, un processus informationnel, une prise de décision, ou une reconnaissance de brevet - ont échoué à capturer l'entière complexité de la pensée de conception. »

Le travail de formalisation de la conception, en une théorie universelle, permettant à chacun de construire une méthode appropriée à ses démarches et problèmes spécifiques, est à poursuivre. Pour ce faire, il convient d'analyser les apports et défaillances des méthodes et théories proposées, et de s'appuyer sur cette compréhension, pour proposer des éléments génériques propres à satisfaire les manques actuels de formalisation. Nous allons maintenant proposer l'étude de quelques-unes de ces méthodes et théories et montrer qu'un des points de formalisation qui reste à développer porte sur la modélisation des problèmes de conception, qui n'a été que peu développée.

Mais, tout d'abord, il nous semble important de situer clairement ce que sont pour nous les notions de conception, de processus de conception, et ce que sont les méthodes et théories de la conception afférentes, afin de clarifier les frontières de notre objet d'étude.

I.1. La conception d'objets techniques, visions et consensus

La conception pose de nombreuses questions et son étude est loin d'être achevée. Acte de l'homme depuis le commencement de l'humanité, elle participe à améliorer son quotidien et à le pousser vers une évolution continue. L'étude de la conception est récente et son approche peut être confuse, considérée comme un acte inné, un processus, des méthodes, une théorie ; tout a été dit sur la conception, mais aucun consensus n'en est ressorti. Nous allons faire le point sur les différentes notions liées à la conception et définir notre cadre d'étude parmi cette multitude.

I.1.1. Définition de la conception

Commençons par rappeler la définition usuelle, donnée par le dictionnaire *Le Petit Robert*² :

1. Formation d'un concept, d'une idée générale dans l'esprit humain.
2. Action de concevoir, acte de l'intelligence, de la pensée, s'appliquant à un objet.
3. Résultat de cette activité intellectuelle, façon de concevoir, ensemble de concepts.

Trois définitions qui portent sur trois notions différentes, à savoir une référence à la généralisation, à la compréhension d'un objet et à la visualisation d'un concept.

Tout d'abord la conception en tant que généralisation. Concevoir réfère à la construction d'une généralisation intellectuelle des objets. Il est donc nécessaire d'abstraire les objets afin de construire des concepts ; dès lors, une question se pose : quels éléments introduire dans la généralisation ? Pourquoi prendre en compte tel constituant d'un objet dans la conceptualisation, plutôt que tel autre ? Nous verrons plus loin que l'un des axiomes de la modélisation est le point de vue (voir [Gartiser et al., 2002]) ; dans le cas de la conception, le point de vue dépend de l'objectif de celle-ci et des connaissances du concepteur. Ainsi la conception est le passage d'un besoin à l'identification d'un concept, orienté par des objectifs à atteindre et limité par les connaissances disponibles, afin de reconnaître, dans ce concept, une solution.

La seconde définition réfère à la compréhension d'un objet, il faut passer d'une idée vague de comportement d'un objet à la compréhension des mécanismes et lois régissant cet objet. Enfin, la troisième définition réfère à la visualisation d'un concept, d'une représentation mentale,

² *Le Petit Robert 1*, par Paul Robert, Dictionnaire alphabétique et analogique de la langue française, rédaction dirigée par A. Rey et J. Rey-Debove, Edition 1989, Dictionnaires Le Robert, Paris.

générale et abstraite d'un objet. Ces deux dernières définitions vont de pair : il faut comprendre pour construire une bonne représentation.

En conception créative de systèmes techniques, il s'agit de passer de l'idée de réalisation d'une fonction, à la compréhension des mécanismes et lois permettant, physiquement, la satisfaction de cette fonction. La conception, dans le cadre de la créativité portant sur des objets techniques, peut alors s'entendre comme la réalisation de choix technologiques. Par exemple, si je souhaite créer une séparation dans un espace, je peux le faire mécaniquement, par la pose d'une cloison, assurant ainsi une isolation thermique, acoustique et physique ; je peux également tendre un ruban, pour signifier la séparation entre deux zones distinctes (zones de sécurité, par exemple) ou peindre, à même le sol, un symbole distinctif (zones de respect devant les guichets) ; je peux également user d'un champ magnétique pour contrôler les autorisations d'accès aux espaces... Il existe de nombreux choix technologiques à valider, en fonction de ce que l'on souhaite réellement séparer, et des ressources à utiliser. Concevoir est donc comprendre les buts, les contraintes et les ressources utiles à la satisfaction d'un objectif. Simon, spécialiste de l'Intelligence Artificielle, considère la conception en tant que synthèse, dans [Simon, 1987] : *« Cela réfère à concevoir des objets, des processus, des idées pour atteindre des buts, et à montrer comment ces objets, processus, ou idées peuvent être réalisés. La conception est un complément de l'analyse - si l'analyse réfère à la compréhension de propriétés et implications d'un objet, processus ou idée qui a déjà été conçue. »*

La norme AFNOR ([NF X 50 127] dans [Le Masson, 2002]) définit la conception comme *« une activité créatrice, qui partant des besoins exprimés et des connaissances existantes aboutit à la définition d'un produit satisfaisant ces besoins et industriellement réalisable. »*

Comme rappelé dans [Mili et al., 2001], l'activité de conception peut être définie pour désigner une grande variété d'activités. Nous l'utiliserons ici pour désigner, spécifiquement, l'ensemble des activités visant à générer et préciser des descriptions détaillées des produits, avant leur réalisation physique. Ce processus de conception industrielle est une démarche analytique, utilisée dans le but d'améliorer la qualité du produit final, ainsi que de réduire le temps et les ressources nécessaires pour la production finale.

I.1.2. La conception en tant que processus

Abordons maintenant un autre aspect de la conception, sa vision en tant que processus. En effet, si nous souhaitons formaliser la conception, il nous faut comprendre les différentes activités propres à cette démarche. Depuis les années 1950, la vision de la conception s'est orientée vers la description d'un processus ([Perrin, 2001]), afin d'augmenter son efficacité et de contrôler son déroulement. Lawson (dans [Lawson et al., 2003]) précise qu'il est aujourd'hui populaire de considérer la conception comme une séquence d'activités, mais que, de plus en plus, il faut considérer non seulement le processus de conception, mais également la pensée créatrice qui le réalise, en s'appuyant sur des méthodes de conception ; nous illustrerons les méthodes de conception plus loin.

Garro définit le processus de conception de la façon suivante dans [Garro, 2000] :

« 1. Le processus de conception n'est pas un processus de résolution de problèmes. La différence tient dans le fait que le problème est souvent co-défini avec la ou les solutions. Lorsque des solutions sont déjà bien connues, le problème va peu évoluer. Mais dans le cas d'innovation, la dérive est généralement importante.

2. Le processus de conception est partiellement irréversible. Ceci veut dire que plus le processus est avancé et plus les retours en arrière sont difficiles car coûteux. Cette irréversibilité est importante économiquement mais aujourd'hui fort mal connue.

3. Le processus de conception est mal défini dans le temps. On ne sait pas précisément quand est-ce qu'il commence et où il s'arrête. »

Ces définitions nous montrent toute l'ambiguïté de la définition du processus de conception. En effet, il dépend de nombreux éléments, ainsi dans [Tate and Nordlund, 1996], le processus de conception est-il défini comme un ensemble d'activités par lesquelles les concepteurs développent et/ou sélectionnent des moyens pour atteindre un ensemble d'objectifs, soumis à des contraintes. Le processus de conception peut donc différer en fonction des moyens disponibles et des objectifs. Tate et Nordlund présentent un modèle du processus de conception (cf. figure I.2.) qui est une généralisation de différentes approches de celui-ci, proposées en Europe ou aux Etats-Unis. La terminologie utilisée par Tate et Nordlund est celle de l'approche axiomatique de la conception (voir [Suh, 1990] et [Suh, 2001]), qui sera présentée ultérieurement. L'intérêt de cette modélisation est de faire la distinction entre les conceptions routinières, basées sur les améliorations de produit, et les conceptions inventives, pour lesquelles un processus de formulation de problème et de génération de concepts est nécessaire.

DP: paramètre de conception
 FR: requis fonctionnels
 C: contrainte
 DM: matrice de conception

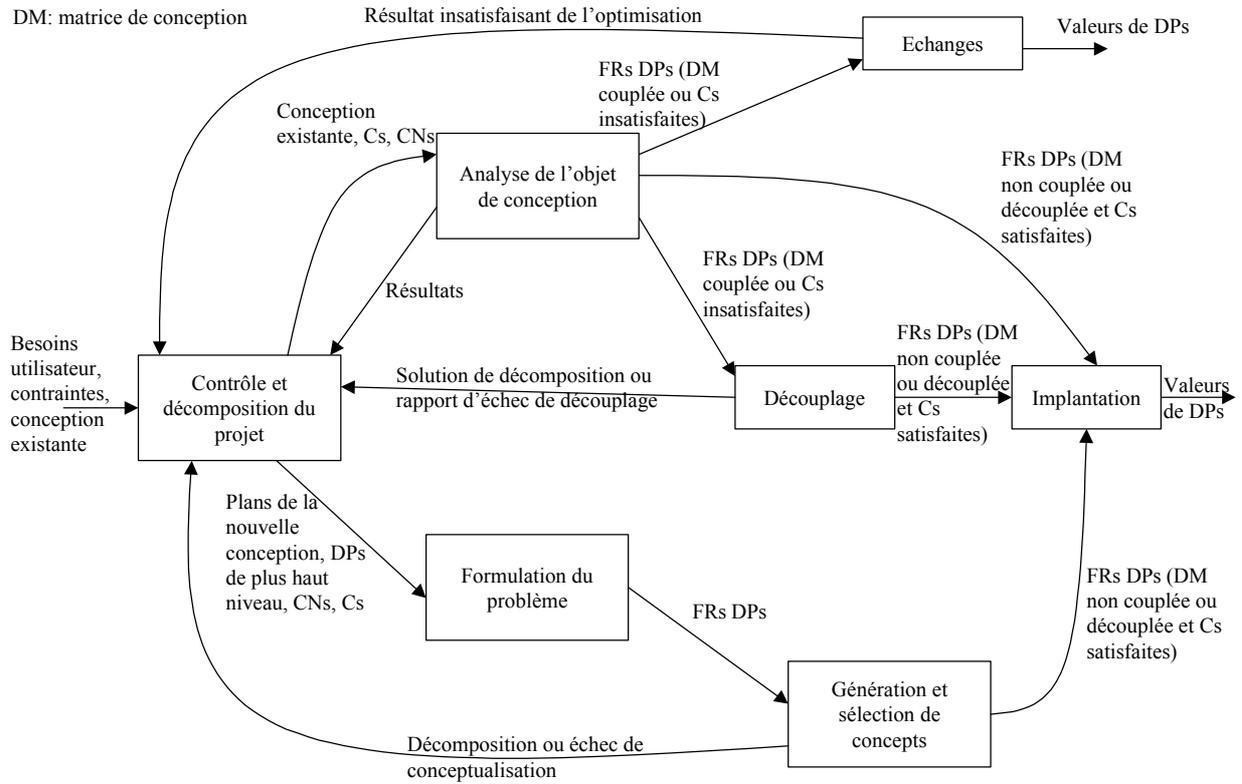


Figure I.2. Modèle du processus de conception, selon [Tate and Nordlund, 1996]

I.1.3. Différentes vues de la conception

Un premier point de distinction des modes de conception est relatif au degré d'inventivité escompté. Le processus de conception diffère, selon qu'on souhaite améliorer la fonctionnalité d'un produit ou créer une innovation de rupture. La figure 1 montre que des attentes, en termes de paramètres de conception, de haut niveau nécessitent une formulation de problème, afin d'identifier les voies technologiques d'obtention de ces paramètres.

Altshuller ([Altshuller, 1988]) a mis en évidence l'existence de cinq niveaux d'inventivité des brevets techniques (voir tableau I.1.).

Niveau	Degré d'inventivité	Proportion	Origine des connaissances
1	Solution apparente	32%	Un individu
2	Amélioration mineure	45%	Une entreprise
3	Amélioration majeure	18%	Une industrie
4	Nouveau concept	4%	Toutes industries
5	Découverte	1%	Ensemble des savoirs

Tableau I.1. Degrés d'inventivité des solutions techniques, selon [Altshuller, 1988]

On constate que la majeure partie des solutions ne réfère qu'à des solutions apparentes, pouvant être résolues par un unique individu, ne nécessitant pas de passer par un processus de résolution de problèmes. De même, Altshuller définit que les améliorations mineures peuvent être apportées par l'utilisation de solutions standards et ne posent pas problème. Ainsi la créativité, en tant qu'apport d'une nouveauté dans un domaine donné (cf. la créativité historique dans [Boden, 1994]), concerne moins de 25% des solutions technologiques ; nous nous intéresserons plus particulièrement, par la suite, aux niveaux 3 et 4 d'inventivité. La découverte est en dehors des préoccupations des entreprises, mais participe plus du domaine de la recherche scientifique, nécessitant d'autres méthodologies. Nous focalisons donc nos travaux sur les processus de conception impliquant des étapes de résolution de problèmes ; ce sont en effet celles-ci qui sont le plus difficilement contrôlées dans les entreprises et qui méritent de gagner en formalisation.

Par ailleurs, dans sa proposition d'une théorie générique pour la conception, Hatchuel revient sur les différentes vues de la conception :

« La conception peut être définie selon les cas comme un acte créateur de mondes ([Goodman, 1978]), une heuristique de résolution de problèmes ([Simon, 1991]) ou un programme systématique ([Pahl and Beitz, 1988]). » [Hatchuel and Weil, 2002]

Nous retrouvons ici la comparaison entre conception et résolution de problèmes. Il est établi que l'activité de conception comporte des phases de résolution de problèmes (voir [Bonnardel, 2000] ou [Gano, 2000]), mais il est également admis que l'une et l'autre ne sont pas en tout point similaires ([Garro, 2000] ou [Visser, 2004]). Ce rapprochement est à la base des travaux en Intelligence Artificielle sur la conception.

Le processus de conception pose donc de nombreuses questions sur la manière de le modéliser. De même, les méthodes de conception, permettant de mener à bien ce processus, sont nombreuses et variées, comme nous allons le voir dans le paragraphe suivant.

I.1.4. Des méthodes de conception variées

« Les méthodes de conception sont des outils et techniques utilisés à différentes étapes de la conception. Toutes les méthodes de conception sont des tentatives pour rendre publique la pensée jusqu'alors privée des concepteurs, afin d'externaliser le processus de conception. L'un des avantages majeurs de rendre la pensée des concepteurs publique est que d'autres personnes, tels que les utilisateurs, peuvent savoir ce qui se passe et fournir des informations et points de vue en plus des connaissances et expérience des concepteurs. » [Sivaloganathan et al., 2001]

Dans cette définition des méthodes de conception, les auteurs insistent sur l'intérêt de la formalisation, en vue du transfert de connaissances et d'une lisibilité de l'activité de conception. Les méthodes permettent d'élaborer des démarches structurées, des étapes clairement identifiables et donc, de justifier les résultats obtenus durant le processus de conception, ainsi que les décisions prises aux diverses étapes du processus.

La définition d'une méthode, selon le Bibliorom Larousse, Version Office 1.0, rappelée dans [Gogu, 2001], indique :

- Ensemble ordonné de manière logique de principes, de règles, d'étapes permettant de parvenir à un résultat ; technique, procédé.
- Ensemble des règles qui permettent l'apprentissage d'une technique, d'une science.

En sus des notions précédentes sur le transfert des connaissances en vue de la lisibilité du processus, ces définitions insistent sur l'intérêt de la formalisation de l'apprentissage. L'apprentissage par la formalisation de méthodes est une approche traditionnelle dans l'enseignement, comme, par exemple, pour les mathématiques. Les méthodes sont en effet des cadres, donnant des repères, lorsque l'on souhaite mener à bien la conception. Le fait de suivre une méthode formalisée ne garantit pas d'obtenir la solution optimale dans un minimum de temps. En effet, suivant la difficulté des problèmes à résoudre et les connaissances du concepteur, un tel pourra être plus rapide sans suivre de méthode. En revanche, une méthode augmente l'efficacité de réalisation du processus de conception, et ce, afin d'obtenir un résultat satisfaisant en un temps limité.

L'intérêt des méthodes étant établi, penchons-nous sur leur caractérisation. S'il en existe d'aussi nombreuses, c'est qu'elles ne sont pas en tout point semblables. Tate, dans [Tate, 1999], propose de classifier les méthodes selon qu'elles sont basées sur les activités de conception ou sur les phases d'évolution de l'objet de la conception. Selon eux, les modèles basés sur les activités s'appuient sur la segmentation suivante :

1. l'analyse : compréhension du problème de conception et génération des requis et spécifications ;
2. la synthèse : génération d'idées et de solutions par l'exploration de l'espace de conception ;
3. l'évaluation : comparaison des solutions aux requis et spécifications ;

tandis que les modèles basés sur les phases se déclinent comme suit :

1. la planification et la clarification de la tâche ;
2. la conception conceptuelle, détermination du principe de solution ;
3. la conception structurelle, spécification de plans successifs ;
4. la conception détaillée, spécification de la production.

Si ces deux approches semblent similaires de par la nature des activités à réaliser, les différences de point de vue peuvent influencer le résultat. Quand une approche centrée sur les activités tend à accroître l'efficacité du processus de conception, en termes de temps passé et de ressources utilisées, une approche centrée sur le produit tend à améliorer la nature de l'objet conçu.

On retrouve une telle distinction dans [Sivaloganathan et al., 2001], travaux classifiant les méthodes selon qu'elles sont descriptives, définissant les activités rencontrées durant le processus, ou prescriptives, par l'apport d'un guide permettant d'améliorer le processus en vue de l'obtention d'un produit optimal.

Enfin, De Araujo, dans [De Araujo, 1996], classe les méthodes de conception en répondant, pour chacune d'elles à trois questions :

- QUAND utilisons-nous telle méthode ?
- QUI utilise telle méthode ?
- COMMENT telle méthode est-elle utilisée ?

Sur cette base, il définit trois niveaux de formalisation des méthodes de conception qui sont :

1. un système d'informations,
2. des connaissances structurées,
3. des outils utilisables.

Afin d'augmenter l'efficacité de l'apport des méthodes, il est nécessaire de tendre vers le niveau de formalisation le plus élevé, à savoir la construction d'outils utilisables.

Les méthodes de conception foisonnent donc, mais toutes ne sont pas égales (voir [Cavallucci, 1999] ou [Vidal, 1999]). Cette profusion n'est pas forcément un bien. Toutes ces méthodes sont comparées dans [Lawson et al., 2003] à des cartes pour guider la conception, mais un regard critique sur leur nombre y est formulé :

« Cependant un danger majeur persiste. Nous en arrivons à penser que ces cartes si abondantes et professionnellement reconnues sont des descriptions précises des pratiques de conception. Le résultat est que la pratique est alors forcée dans un cadre duquel on ne peut sortir et qui ne conduit pas forcément à de meilleures conceptions. »

L'aspect rigide de ces méthodes est présenté tel un frein à leur utilisation. Enfin nous pouvons noter que, toutes étant spécifiques, il n'en est aucune de générique, qui serait adaptée à tout concepteur dans toute industrie. La question se pose d'ailleurs : est-il possible de proposer une méthode de conception adaptée à tous ? C'est là un fameux programme de conception ! Les éléments de réponse, apportés aujourd'hui, tendent à montrer qu'il vaut mieux compléter et intégrer spécifiquement des parties de méthodes, afin d'accroître l'efficacité du processus de conception (cf. [Tate and Nordlund, 1995] et [Cavallucci et al., 2000]). Il s'agirait donc de développer des outils spécifiques pour chaque phase du processus de conception, outils pouvant s'intégrer à toute méthodologie existante en entreprise.

I.1.5. Positionnement vis-à-vis de la conception

Nous souhaitons concentrer nos travaux en conception de systèmes techniques sur la phase de conception conceptuelle. Il s'agit de la phase durant laquelle les concepts de solution sont produits. Il nous semble important de répondre à une attente, fortement ressentie dans le monde industriel, en termes d'outillages pour la génération des concepts. Sans réduire pour autant la conception de produits à la seule génération des concepts, mais l'ensemble du processus est largement outillé et relativement bien maîtrisé aujourd'hui, au regard de cette phase.

Notre approche est de considérer la conception conceptuelle comme un processus de résolution de problème. Nous reconnaissons deux avantages à cela :

1. la vision du processus cognitif de formation de concepts en tant que modifications successives de modèles ;
2. l'identification des concepts par l'utilisation d'analogies.

La vision du processus cognitif en tant que modifications successives de modèles est largement répandue dans la résolution de problèmes. Antonietti rappelle, dans [Antonietti et al., 2000], les cinq perspectives théoriques majeures sur lesquelles sont construites les méthodes de résolution de problème :

- la production massive d'idées, le brainstorming ([Osborn, 1953]) en est une illustration usitée en conception ;
- la recherche de solution par combinaisons et arrangement d'éléments séparés, l'analyse morphologique ou les travaux de De Bono ([De Bono, 1976]) sont basés sur ce principe ;
- l'utilisation des analogies, la mise en évidence de correspondances entre des situations issues de domaines différents. La synectique de Gordon ([Gordon, 1961]) et les travaux d'Altshuller ([Altshuller, 1988]) fonctionnent sur le principe de l'analogie ;
- la vision du processus de résolution de problème comme une restructuration de la situation rencontrée. Cela impose de considérer le problème dans son ensemble, d'en comprendre ses caractéristiques principales et de tenter de la percevoir sous différents points de vue. Le changement de point de vue est l'un des principes de description de la situation de conception dans OTSM

([Khomenko and Sokol, 2000]) et dans l'analyse fonctionnelle ([NF X 50 100]).

- l'application d'opérateurs permettant la transformation de la situation initiale du problème en un état de résolution. La décomposition des tâches, objectifs, en sous-tâches, sous-objectifs en est un exemple.

Toutes ces visions partagent le point commun d'une évolution d'états entre un état initial, représentation de la situation problématique, et un état final, représentation de la solution ou concept de solution (cf. [Stefik, 1995]). Les différences se font sur le niveau de généralité des états construits et sur les modes d'évolution entre deux états consécutifs.

Nous avons vu que les approches de la conception en tant que processus de résolution de problèmes - à travers les exemples de la synectique de Gordon, de la TRIZ, de l'analyse fonctionnelle, du brainstorming - sont courantes. Pourtant elles ne nous semblent pas explicitées clairement dans les méthodes de conception.

Nous allons voir, par la description d'un certain nombre de méthodes de conception usuelles, la présence de la résolution de problème et la formalisation de celle-ci, en axant notre vue sur les phases de conception conceptuelle. Cette description se propose de montrer la présence de modes de représentation et de résolution de problèmes dans les approches de conception, et de mettre en évidence l'absence de cadre de formulation permettant une démarche globale de résolution de problèmes.

1.2. Les méthodes traditionnelles de la conception

Nous allons détailler dans ce paragraphe les méthodes et théories les plus répandues dans le milieu industriel. Notre objectif est ici de décrire, à la fois la prise en compte du processus de résolution de problèmes, et le degré de formalisation de ce processus.

1.2.1. L'approche systématique de Pahl et Beitz

1.2.1.1. Une approche prescriptive convergente

L'approche systématique de la conception, proposée par Gerhard Pahl et Wolfgang Beitz ([Pahl and Beitz, 1988]), repose sur l'étude de la psychologie, de la philosophie et des sciences des facteurs humains. Elle apporte une représentation du processus de conception comme succession hiérarchique de phases. Perrin ([Perrin, 2001]) décrit le modèle de Pahl et Beitz comme un modèle prescriptif, c'est à dire un modèle qui tente de prescrire un patron d'activités meilleur et plus approprié ([Sivaloganathan et al., 2001]). Par ailleurs au cours du processus de conception, le recueil des informations tend à réduire les alternatives de solutions, le processus est donc convergent.

La décomposition du processus de conception (cf. figure I.3.) repose sur les principales étapes de planification et de clarification des tâches (spécification des informations), la conception conceptuelle (définition des principes), l'expression de la conception (définition des plans) et la conception détaillée (définition de la production).

La première phase, de planification et de clarification de la tâche, permet d'obtenir une description des objectifs de l'étude basée sur les connexions entre les fonctions du produit, ses conditions d'obtention et de réalisation. L'attention est portée sur la description du produit à réaliser en termes de fonctions désirées, de coûts, de requis préliminaires. Pour clarifier la tâche, une collecte d'informations est initiée dès cette phase, afin de comprendre de quoi traite réellement le problème : « Quels souhaits ou attentes non exprimées doivent être pris en compte ? », « Quelles sont les contraintes liées au problème ? ». Cette première phase de conception correspond à l'établissement du cahier des charges, soit les spécifications techniques et économiques à atteindre.

Dans la phase de conception conceptuelle, un concept de solution est formulé sur la base de l'analyse fonctionnelle et des alternatives disponibles, selon un schéma de résolution de problèmes. La conception structurelle permet de définir les formes et les dimensions de la

solution. Enfin la phase de conception détaillée est nécessaire pour l'élaboration des documents de fabrication du produit.

Il faut noter que le processus n'est pas forcément linéaire, mais des itérations peuvent être nécessaires durant le déroulement du processus.

L'approche proposée par Pahl & Beitz est un cadre d'opérations à effectuer dans le but de concevoir avec efficacité une solution technique. La modélisation repose sur une description fonctionnelle et économique du produit, par la constitution d'un cahier des charges. Schématiquement, l'approche décrit les actions à réaliser mais ne formalise pas les démarches de réalisation de ces actions.

La phase de conception conceptuelle aborde la résolution de problèmes, sur laquelle elle s'appuie pour réaliser la transition entre un cahier des charges et des concepts de solutions satisfaisant ce cahier des charges. L'un des éléments de base de ce passage est l'abstraction. Nous allons détailler la phase conceptuelle dans la partie suivante.

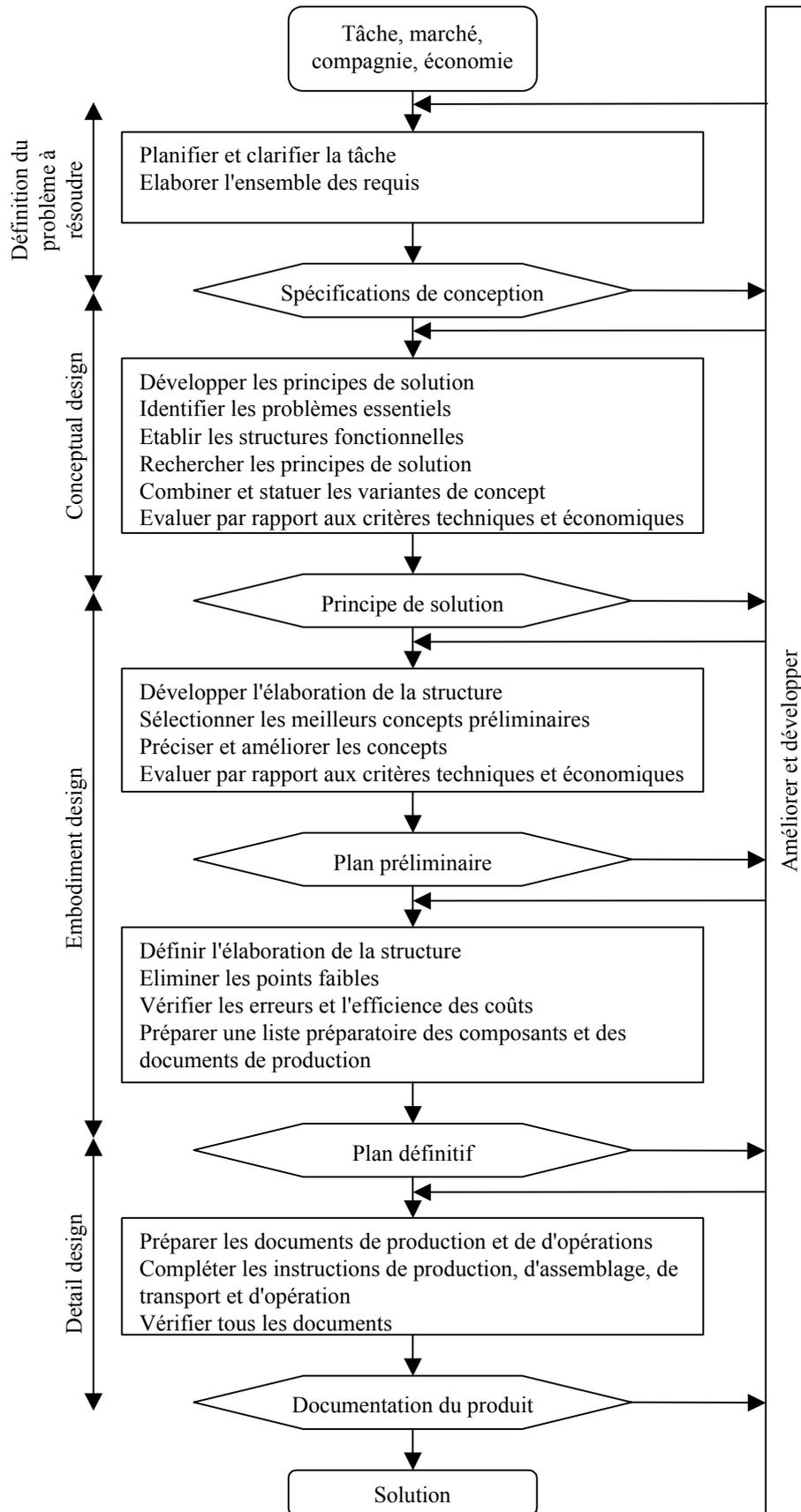


Figure 1.3. Les étapes du processus de conception, selon [Pahl and Beitz, 1988]

I.2.1.2. La phase de conception conceptuelle, un processus de résolution de problèmes

La conception conceptuelle est celle durant laquelle sont identifiés et résolus, grâce à une phase d'abstraction, les problèmes clés. L'identification des problèmes se construit sur l'établissement de la structure fonctionnelle, la recherche de solutions de principe et de leur combinaison (cf. figure I.4.).

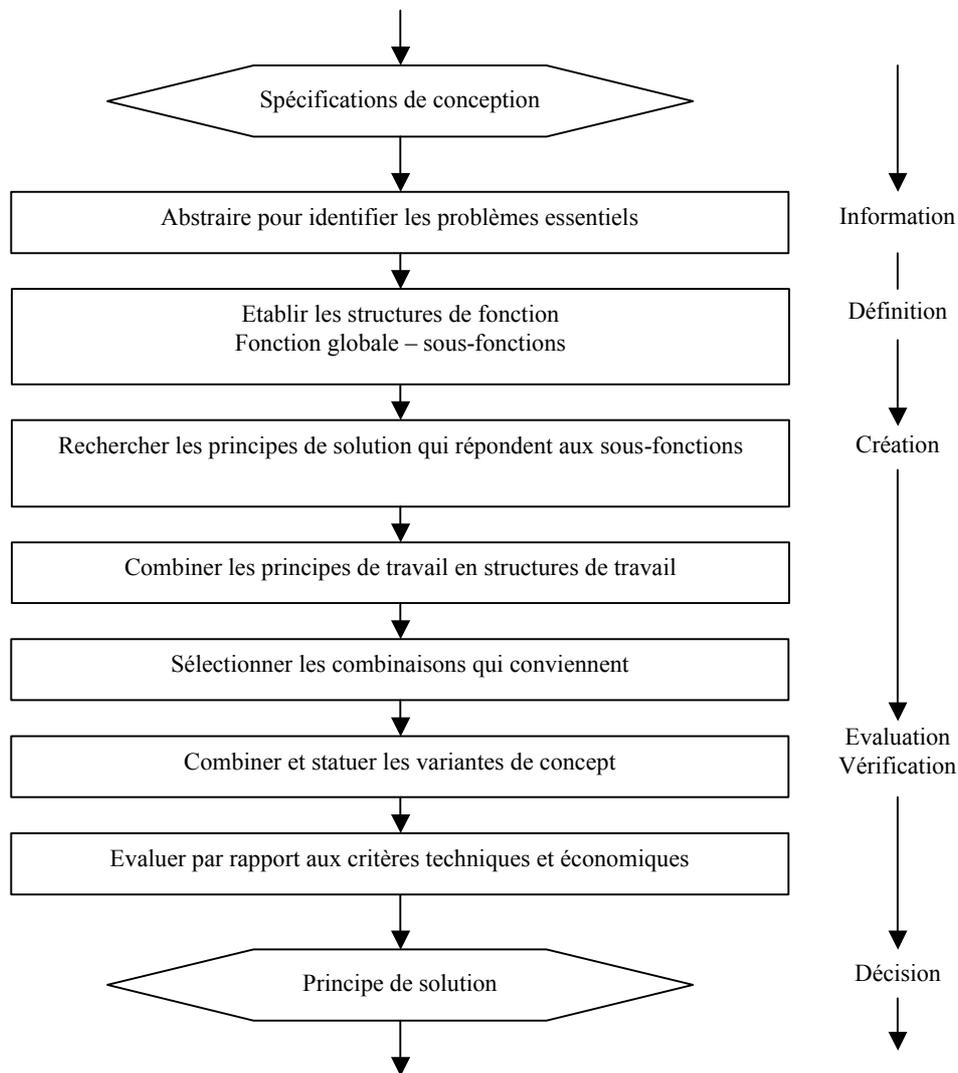


Figure I.4. Les étapes de la conception conceptuelle, selon [Pahl and Beitz, 1988]

Dans la recherche de la solution optimale, les concepteurs peuvent être influencés par des idées figées et conventionnelles. Pour dépasser cette fixation, Pahl et Beitz préconisent l'utilisation de l'abstraction ; elle permet de générer des concepts se détachant du domaine d'application, en ne s'attachant qu'à la description des caractéristiques requises pour la solution. Ceci implique de se détacher de ce qui est spécifique au problème, de relever les points génériques et essentiels du problème. Une telle généralisation conduit à l'identification du cœur

de la tâche. Une bonne formulation générique dégage clairement la fonction principale et les contraintes essentielles sans pour autant nuire au choix d'une solution particulière. L'utilisation de l'abstraction, accompagné de l'établissement de la liste des requis, permet l'identification des caractéristiques principales du futur produit à concevoir. Pahl et Beitz proposent l'algorithme suivant pour la conduite de cette analyse :

1. éliminer les préférences personnelles,
2. omettre les exigences n'ayant pas de lien direct avec la fonction et les contraintes essentielles,
3. transformer les éléments quantitatifs en données qualitatives,
4. généraliser les résultats des étapes précédentes,
5. formuler le problème avec des termes non liés à une solution.

La recherche d'un principe de solution est basée, dans un premier temps, sur la satisfaction d'une fonction technique.

La conception conceptuelle permet d'aboutir à une formalisation générique du problème et fournit des solutions pouvant se révéler totalement nouvelles pour le concepteur qui, par le biais de l'abstraction, doit explorer des voies hors de son domaine de compétences.

Pahl et Beitz proposent une approche pas à pas de la conception. La méthode s'appuie fortement sur l'aptitude à l'abstraction, la systématisation des tâches et la pensée logique comme compléments aux connaissances professionnelles des concepteurs.

1.2.1.3. Un cadre prenant en compte la résolution de problèmes, une démarche non outillée

L'apport principal de l'approche systématique est de proposer une démarche méthodique et structurée de la conception. L'approche systématique fournit un cadre méthodologique relativement large, en tant que modèle du processus de conception ([Malmquist et al., 1996]). Par ailleurs, la systématisation de la démarche permet la capitalisation de certains savoirs métiers afin d'élaborer des catalogues de conception, constituant ainsi une base considérable pour aider à faciliter la démarche de conception.

La vision du processus de résolution de problèmes est intégrée aux phases de conception conceptuelle. Par contre, la constitution progressive des données du problème n'est pas totalement formalisée, même si les retours en arrière préconisés sont une manière d'y répondre. Nous pensons que la capitalisation des connaissances relatives au problème se fait tout au long

du processus de conception, formant ainsi un processus parallèle. La formulation du problème doit donc permettre d'intégrer toute connaissance apparaissant à un moment donné du processus de conception.

Ainsi le passage de la spécification des tâches et caractéristiques à la formalisation de concepts doit être un processus progressif, et itératif. Si cette vision est sous-jacente à l'approche de Pahl & Beitz, elle n'y est pas clairement explicite. Quels sont toutes les informations à recueillir au cours de ce processus ? Quel modèle de représentation des connaissances construire ? Pahl & Beitz se réfèrent à la constitution d'un cahier des charges et mettent en avant l'importance de la description fonctionnelle des besoins, nous allons décrire l'analyse de la valeur qui est basée sur ces principes et offre une formalisation plus avancée des éléments à capitaliser.

I.2.2. L'analyse de la Valeur, une approche centrée sur les fonctions

I.2.2.1. Une démarche qualitative pluridisciplinaire

« L'analyse de la Valeur est une démarche créative et organisée utilisant un processus de conception fonctionnel et économique, dont le but est d'augmenter la valeur d'un sujet AV. »
([NF X 50 150-1])

L'analyse de la valeur est basée sur les notions primordiales de fonctions et de ressources. Cette méthode, mise en application par Lawrence D. Miles en 1947, vise à rendre le produit conforme à ce que le client en attend (optimisation des fonctions), en réduisant au strict nécessaire les ressources employées. Les ressources sont les moyens mis en œuvre pour atteindre les objectifs, exprimés en termes de fonctions à rendre par le produit.

La norme [NF X 50 152] définit le processus d'une étude d'analyse de la valeur en sept phases :

1. Orientation de l'action Analyse de la Valeur : définition de l'objet, des données techniques et fonctionnelles existantes, des limites, contraintes, ...
2. Recherche de l'information : inventaire des informations nécessaires.
3. Analyse des fonctions et des coûts, validation des besoins et des objectifs : réalisation du Cahier des Charges Fonctionnel, répartition des coûts par fonction.
4. Recherche d'idées et de voies de solution : utilisation des techniques de créativité.
5. Etude et évaluation des solutions : études techniques de faisabilité, prise en compte des coûts, risques et contraintes.
6. Bilan prévisionnel, présentation des solutions retenues, décision.
7. Réalisation, suivi, bilan.

L'Agence Française de l'Analyse de la Valeur ([AFAV, 1994]) définit l'analyse de la valeur comme une démarche :

- *« fonctionnelle, ce qui impose de formuler les problèmes en termes de finalités et non en termes de solutions, pour rechercher l'essentiel et le pertinent*
- *à caractère économique, par la référence systématique aux coûts*

- *pluridisciplinaire, par un travail de groupe réunissant toutes les compétences requises, et mettant en présence des personnes de formation et de responsabilités différentes. Ce qui permet de trouver un consensus sur les fonctions, les performances, les principes, les solutions, les coûts. Ce travail de groupe permet de régler conjointement des problèmes qui, sinon, ne seraient abordés que successivement et isolément par les divers intervenants dans la création et la réalisation du produit. »*

L'analyse de la valeur s'appuie sur l'Analyse Fonctionnelle pour exprimer fonctionnellement le besoin, c'est un outil rigoureux qui a pour mission de considérer un produit comme un ensemble de fonctions ([Duchamp, 1988]). Aujourd'hui développée en tant que démarche autonome l'analyse fonctionnelle conduit à l'expression fonctionnelle du besoin et au cahier des charges fonctionnel. On peut considérer que l'analyse fonctionnelle permet la construction de l'état initial de représentation du problème par l'analyse exhaustive de la situation. Par ailleurs l'identification des finalités, des fonctionnalités à réaliser posent les caractéristiques permettant de reconnaître un état comme concept de solution. La construction du problème, la représentation de la situation problématique initiale, et la description des objectifs à atteindre (des caractéristiques de la solution), sont formalisées dans l'analyse fonctionnelle.

1.2.2.2. L'Analyse Fonctionnelle propose un langage de modélisation

« L'analyse fonctionnelle est une démarche qui décrit complètement les fonctions et leurs relations, qui sont systématiquement caractérisées, classées et évaluées. » ([NF X 50 100])

L'analyse fonctionnelle fournit un cadre de représentation des finalités d'une étude. Elle apporte un mode de construction de la connaissance utile en proposant une sémantique précise de caractérisation de la situation problématique et des objectifs. La situation est définie par l'environnement d'utilisation et la prise en compte de l'utilisateur, les objectifs sont décrits par une classification de fonctions.

Il nous paraît intéressant de rappeler ici certaines notions définies par la norme [NF X 50 150], et relatives à la définition claire des problèmes à résoudre dans le processus de conception :

- *« Le **besoin** : c'est ce dont un utilisateur éprouve la nécessité ou ressent le désir. Le besoin est l'expression des attentes, d'un désir éprouvé par un individu, un groupe d'individus. Il peut être explicite (et donc exprimé) ou implicite, latent ou potentiel, avoué ou inavoué, persistant ou éphémère.*

- Les **fonctions** : le concept de fonction est le concept de base de l'Analyse fonctionnelle. Une fonction est définie comme l'action d'un produit ou de l'un de ses constituants exprimés uniquement en termes de finalité, pour son effet recherché. Le concept peut être décliné sur l'ensemble du cycle de vie du produit. Les **fonctions de service** sont celles que doit satisfaire le produit pour répondre aux besoins de l'utilisateur. La classification actuelle de la norme distingue parmi ces fonctions de service, les **fonctions d'usage** (qui traduisent la partie rationnelle du besoin) et les **fonctions d'estime** (qui traduisent la partie subjective du besoin). Une deuxième classification propose de distinguer dans les fonctions de service les **fonctions principales** qui correspondent aux services rendus par le produit pour répondre au besoin de l'utilisateur et les **fonctions complémentaires** qui traduisent des réactions, des résistances ou des adaptations à des éléments du milieu extérieur. Les **fonctions techniques** représentent les actions internes au produit (entre ses constituants), dans le cadre d'une solution, et qui sont nécessaires pour satisfaire les fonctions de service. »

L'analyse fonctionnelle a pour but d'assurer une bonne formulation de l'expression du besoin en termes de fonctionnalités à rendre par le produit, en outre elle amène :

- une hiérarchisation des fonctions à assurer par ordre d'importance,
- l'expression de critères d'appréciation de la fonction et la caractérisation des niveaux attribuables à ces critères d'appréciation,
- la formulation de la flexibilité qui caractérise ce que le demandeur accepte de voir modifier sur les performances du futur produit.

L'analyse fonctionnelle propose une modélisation du problème par la pose d'un cadre de formulation et de caractérisation des fonctions. La base de cette construction se définit dans un document de travail dont l'importance est reconnue : le Cahier des Charges Fonctionnel.

1.2.2.3. Le Cahier des Charges Fonctionnel, un outil de caractérisation de la solution

« Le Cahier des Charges Fonctionnel est un document par lequel le demandeur exprime son besoin (ou celui qu'il est chargé de traduire) en termes de fonctions de service et de contraintes. Pour chacune d'elles sont définis des critères d'appréciation et leurs niveaux. Chacun de ces niveaux doit être assorti d'une flexibilité. » ([NF X 50 150])

Cette définition introduit un concept jusqu'alors non mentionné : celui de contrainte. La norme [NF X 50 100] définit la contrainte comme une « *caractéristique, effet ou disposition de conception, qui est rendue obligatoire ou a été interdite pour quelque raison que ce soit. Aucune autre possibilité n'est laissée.* » Cette notion de contrainte demeure toutefois ambiguë, dans le mode de formalisation, d'autant que sont affectés aux contraintes des critères d'appréciation et des niveaux ([Prudhomme, 1999]). Ainsi fonctions et contraintes prennent un statut similaire, pourquoi dès lors ne pas considérer de fonction contrainte ? Si on le faisait (et c'était d'ailleurs initialement le cas, voir [Galisson, 1996]), il existerait un cadre de formulation des contraintes, de même que pour les fonctions.

Le Cahier des Charges Fonctionnel peut donc être vu comme un recueil permettant de positionner les limites et points clés des produits à concevoir. En effet le document comprend ([Prudhomme, 1999]) :

- « **La présentation du problème** : elle doit inclure la présentation du produit (concepts du produit, services attendus) et de son marché (débouchés prévus, ...), le contexte du projet et les objectifs visés (limites d'étude, études déjà effectuées, suites prévues, ...).
- **L'énoncé fonctionnel du besoin** : qui est le résultat d'une analyse fonctionnelle du besoin. Cependant on retrouve l'ambiguïté précédente puisque les contraintes sont mises au même niveau que les fonctions dans cet énoncé fonctionnel du besoin. Pour chacune d'elles (FS et contraintes) doivent être précisés les critères d'appréciation, niveau, flexibilité. La flexibilité permet d'organiser le dialogue entre partenaires dans la recherche d'une optimisation (entre performance, coût, délai). Bien sûr, le produit devra satisfaire à la somme des fonctionnalités identifiées en tenant compte de l'ensemble du cycle de vie du produit. Les contraintes sont-elles vues comme une fonctionnalité ?
- **Un appel à variantes** : il peut être demandé au concepteur (à voir comme le concepteur-dessinateur) de fournir une proposition répondant à sa propre perception du besoin (variantes fonctionnelles ou performanciennes) en plus d'une proposition répondant à l'expression fonctionnelle du besoin. Pour la norme, cette approche stimule l'innovation tout en permettant un dialogue entre demandeur et concepteur.

Un cadre de réponse : pour faciliter le dépouillement des réponses, un cadre doit être prévu pour porter toutes les précisions liées à la solution proposée. Ce cadre concerne chaque fonction, mais aussi l'ensemble du produit. Les réponses obtenues engagent le concepteur et constituent une amorce de spécification de réception. » Le Cahier des Charges Fonctionnel pose clairement la situation initiale, ainsi que les caractéristiques permettant de reconnaître une situation finale satisfaisante. En revanche, le cheminement permettant de passer de l'une à l'autre n'est pas explicité, et encore moins formalisé. Ce cheminement est censé s'appuyer sur la résolution de problèmes. Or, la spécification de la situation initiale n'est pas la modélisation d'un problème, dans le sens où cette spécification ne repose pas sur la mise en évidence des obstacles limitant la réalisation des fonctions souhaitées.

I.2.2.4. Contribution à une approche de la conception par la résolution de problèmes

L'analyse de la valeur, notamment les étapes de l'analyse fonctionnelle, permettent de positionner très clairement une étude de conception. La définition des objectifs de l'étude, des limites donnent des critères de validation des solutions proposées et sont autant d'étapes nécessaires à toute étude de conception. En outre, l'analyse de la valeur propose un ensemble d'outils pour la réalisation de chaque étape, le niveau de formalisation globale est donc des plus intéressants. L'approche fonctions/coûts pose les bases d'une vision duale de la conception des systèmes techniques, avec une face fonction-valeurs et une face artefacts-coûts (cf. figure I.5.).

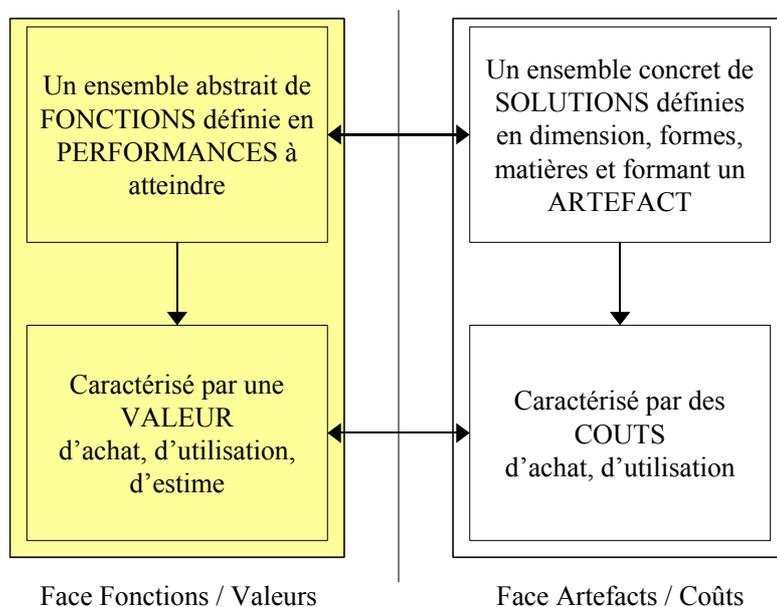


Figure I.5. Les deux faces d'un produit, selon [BTE, 1991]

Ces deux faces de la conception se retrouveront dans toutes les approches qui suivront, notamment dans la théorie de la conception axiomatique (voir chapitre I.2.3.).

Si le niveau de formalisation et de systématisation des tâches de l'analyse de la valeur est intéressant, nous pouvons noter qu'il existe une hétérogénéité de cette formalisation. En effet, il n'existe pas d'outils, ni même de structuration des connaissances (second niveau de formalisation de [De Araujo, 1996]), pour la phase de génération des concepts de solution (cf. [Cavallucci, 1999]). La démarche amène le concepteur à positionner son objet d'étude, à identifier les caractéristiques clés, les contraintes à respecter ; la phase de génération des solutions repose alors totalement sur la créativité de l'individu. S'il fallait accroître le niveau de formalisation de l'analyse de la valeur, c'est certainement sur ce point que nous devrions porter notre attention.

Par ailleurs [Yannou et al., 2002] pointe le fait que l'analyse de la valeur ne prend en compte ni les incertitudes ni les risques et se centre plus sur la modélisation du produit que sur celle du projet.

L'analyse fonctionnelle pose le problème de la transition des fonctions aux éléments de réalisation de ces fonctions. Le passage est présenté ici comme une phase de créativité et aucun guide de réalisation de celle-ci n'est fournie. En outre, la décomposition fonctionnelle semble pouvoir se faire indépendamment de choix technologiques, or il paraît suspect de lister les sous-fonction sans connaître la technologie associée à la fonction. Suh apporte une réponse à cette problématique dans l'Axiomatic Design.

I.2.3. L'approche axiomatique de la conception

I.2.3.1. Objectifs de l'approche axiomatique de la conception

[Kurr, 1998] nous rappelle que Suh, inventeur de l'Axiomatic Design, a fait de l'enseignement de la conception sa préoccupation première lors du développement de cette approche. Suh a cherché à établir les principes fondamentaux et les méthodes qui guident la prise de décision durant le processus de conception ([Suh et al., 1978]). L'objectif de l'Axiomatic Design est d'établir une base scientifique de la conception et d'améliorer les activités de conception par l'apport aux concepteurs de fondations théoriques basées sur l'étude des processus mentaux et la proposition d'outils rationnels et logiques ([Suh, 2001]). Une définition de la conception est donnée dans [Suh, 1990] : « La conception, qui est le cœur même de l'ingénierie, facilite la création de nouveaux produits, procédés, logiciels, systèmes et organisations par laquelle l'ingénierie contribue à la société en satisfaisant ses besoins et aspirations. » On voit apparaître, dans cette définition, la notion de besoin qui est primordiale dans l'approche axiomatique, et qui relève de la modélisation des problèmes.

Le développement des connaissances doit débiter par la formalisation d'axiomes qui sont ensuite développés sous forme d'algorithmes, pour être finalement proposés en outils. Une approche axiomatique part du prémisses qu'il existe des principes généralisables qui guident le comportement du système étudié, ici la conception ([Suh, 2001]). La conception est un échange itératif entre ce que nous souhaitons réaliser et comment nous choisissons de satisfaire ce besoin (cf. figure I.3.). Pour systématiser le processus de pensée au cours de cet échange, le concept de domaines créant des démarcations entre quatre types d'activités de conception est une base importante de la conception axiomatique. Ainsi, la conception peut être décrite comme un processus permettant de passer à travers quatre domaines. Ces domaines représentent différents points de vue du besoin, des points de vue utiles à une construction progressive de la solution. Les quatre domaines sont ceux montrés sur la figure I.6., à savoir :

- le domaine client est caractérisé par les besoins que le client souhaite voir réaliser, les attributs du client, {CA}, y sont exprimés ;
- le domaine fonctionnel, dans lequel les CAs sont traduits par les fonctions que le système à concevoir aura à satisfaire {FR} et par un ensemble de contraintes {C} ;
- le domaine physique permet d'identifier les paramètres physiques qui permettent la réalisation des fonctions {DP} ;

- le domaines des procédés enfin, permet d'exprimer les variables des processus de conception qui réalisent les paramètres physiques {PV}.

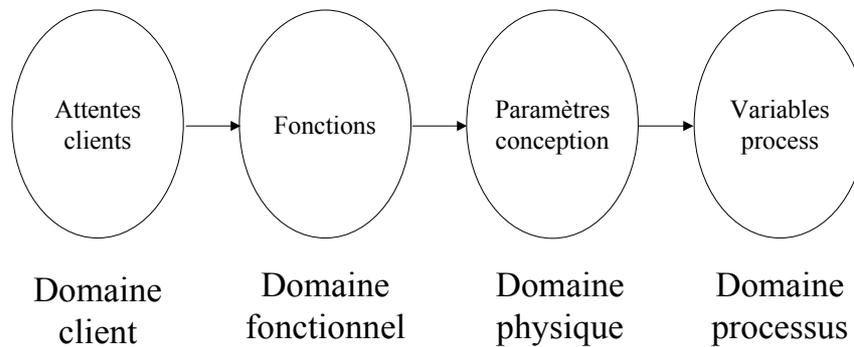


Figure I.6. Les quatre domaines de la conception, selon [Suh, 2001]

Une fois ces domaines identifiés, il nous faut comprendre comment le concepteur les traverse, réalisant ainsi le processus de conception.

1.2.3.2. La conception, un parcours à travers quatre domaines

L'axiomatic Design a décrit de façon précise ces ensembles et notamment les relations liant ces ensembles entre eux. Ces relations permettent de réaliser le passage entre les domaines. La méthode préconisée par Suh peut être décrite de la manière suivante ([Dubois and Lutz, 2003a]) :

1. Lister les attentes du client, celles-ci peuvent être parfois difficiles à identifier et se révéler relativement vagues.
2. Transformer les CAs en un ensemble minimal de requis fonctionnels (FRs) indépendants permettant la satisfaction de ces attentes.
3. Lister les contraintes (Cs) auxquelles le système devra obéir.
4. Identifier les paramètres physiques de conception qui réalisent les fonctions, par un zig-zagging entre les domaines fonctionnel et physique (cf. figure I.7.).
5. Valider la conception par la vérification de l'axiome d'indépendance et les contraintes.
6. Identifier les variables des processus de conception pour chaque DP.

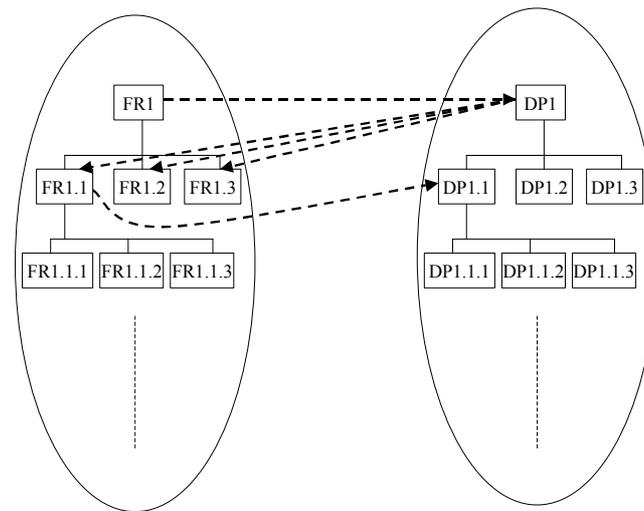


Figure I.7. Le zig-zagging entre les domaines fonctionnel et physique

Il nous semble important de définir certains concepts de la méthodologie, utiles à la compréhension de la construction d'un modèle de problème. La première de ces notions est celle de contrainte. Suh définit les contraintes comme les limites acceptables des solutions. Il identifie deux types de contraintes :

- Les contraintes d'entrées qui sont imposées comme spécifications de la conception.
- Les contraintes du système imposées par le systèmes dans lequel la solution devra évoluer.

Il est important de préciser également le concept des zig-zagging. Cette notion exprime le passage des requis fonctionnels aux paramètres de conception. L'intérêt majeur de cette approche est de définir que les choix de technologies influent sur la construction hiérarchique des fonctions. Ainsi, il n'est pas possible de définir l'ensemble des fonctions qu'un système aura à réaliser, tant que les choix technologiques ne sont pas réalisés. Cette description est une réelle avancée sur l'analyse fonctionnelle.

Il est en revanche étrange de constater la non-intégration des contraintes au zig-zagging. L'identification des paramètres physiques de conception reste donc sujette à la satisfaction des contraintes formulées.

Les choix technologiques se font au cours de la description fonctionnelle, la décomposition permet d'identifier pour chaque « Quoi ? » un « Comment ? ». La question qui subsiste alors est : « Comment sait-on que les bons choix technologiques ont été faits ? ». Pour y répondre Suh propose l'axiome d'indépendance.

1.2.3.3. Un critère de classification des solutions

Les requis fonctionnels (FRs) doivent être définis comme l'ensemble minimal des fonctions indépendantes à réaliser. L'axiome d'indépendance identifie une solution acceptable comme une solution pour laquelle les paramètres de conception et les requis fonctionnels sont reliés de manière à ce qu'un paramètre de conception spécifique puisse être ajusté de manière à satisfaire le requis correspondant sans affecter les autres FRs ([Tate and Nordlund, 1995]).

Suh exprime cette relation d'indépendance sous une forme mathématique. A un niveau hiérarchique donné, l'ensemble des FRs forme un vecteur dans le domaine fonctionnel. De même les paramètres de conception à identifier constituent un vecteur dans le domaine physique. La relation entre ces deux vecteurs est exprimée par l'équation (1.1) :

$$\{FR\} = [A]\{DP\} \quad (1.1)$$

La matrice [A] est appelée la matrice de conception, elle caractérise la qualité de la conception du produit, en termes d'indépendances des fonctions. Cette matrice permet de classer les conceptions, suivant la forme de la matrice. Trois cas de figure sont identifiés :

- Conception non couplée : si la matrice [A] est diagonale, chaque FR peut être satisfait indépendamment au moyen d'un DP.
- Conception découplée : si la matrice [A] est triangulaire, l'indépendance des FRs peut être assurée par la détermination des DPs dans une séquence spécifique.
- Conception couplée : dans tout autre cas, l'indépendance des FRs n'est pas satisfaite, la conception est alors à revoir.

Cette indépendance des FRs doit être vérifiée à tout niveau hiérarchique, durant tout le processus de zig-zagging.

1.2.3.4. Un critère de faisabilité

Le second axiome permet d'identifier parmi plusieurs solutions possibles, laquelle correspond à la conception optimale. L'axiome d'information statue que la conception ayant la plus haute probabilité de succès est optimale. Pour chaque FR_i un contenu d'information I_i peut être défini par l'équation 1.2, où P_i est la probabilité de satisfaire FR_i :

$$I_i = \log_2 (1/P_i) = -\log_2(P_i) \quad (1.2)$$

Le second axiome peut alors s'exprimer par la minimisation du contenu d'information de la conception. La meilleure conception est celle qui possède le contenu d'information le plus bas, car elle requière une quantité d'information minimale pour la satisfaction des objectifs de la conception.

Une solution est alors dite complexe si elle possède une faible probabilité de succès, soit un contenu d'information élevé. Cela se produit si l'on a des tolérances faibles sur les FRs pour un produit (ou sur les DPs pour un procédé), nécessitant une précision importante.

L'axiome d'information est un outil performant pour choisir parmi différents ensembles de paramètres de conception.

1.2.3.5. Une théorie sans méthode

L'un des apports fondamentaux de l'approche proposée par Suh est de proposer une description générique de la conception, jetant ainsi les bases d'une théorie de la conception. En outre la description du processus de zig-zagging nous semble d'un intérêt réel et répond à l'un des manques de l'analyse fonctionnelle qui ne représente pas cette nécessaire interaction entre formalisation fonctionnelle et choix technologiques.

Les axiomes, d'indépendance et d'information, permettent de réaliser les choix lorsque plusieurs voies de réalisation des fonctions se présentent ; la description analytique des axiomes apporte aux concepts générés une validité mathématique (voir [Rudolph, 1996]). Il est noté dans [Harutunian et al., 1996] qu'une telle validité permet le développement d'outils informatiques pour systématiser les tâches. Toutefois une décision prise à un niveau hiérarchique peut-elle assurer la continuité d'optimum pour les étages suivants ? Cette question reste aujourd'hui ouverte.

La limitation principale de l'approche axiomatique est qu'il n'existe pas de méthode pour identifier, à partir de la formalisation d'un FR, le DP correspondant. Le passage a besoin d'être outillé afin d'optimiser le processus de conception (cf. [Dubois and Lutz, 2003b]). Les éléments principaux de modélisation de problèmes sont présents, de par l'identification des fonctions à réaliser, des contraintes à satisfaire et par la construction progressive de la solution. En revanche, la résolution des problèmes n'est pas intégrée, pourtant le passage du domaine fonctionnel au domaine physique relève d'un processus de résolution de problèmes, pour la phase de conception conceptuelle.

I.2.4. Le Quality Function Deployment

I.2.4.1. Présentation de la démarche

Le développement de la fonction qualité (QFD) est :

« Un concept global qui fournit les moyens de traduire les besoins du client en requis techniques appropriés à chaque étape du développement et de la production du produit. » ([Sullivan, 1986]).

Cette définition montre la globalité de l'approche du QFD, qui est appliquée à différentes étapes du processus de conception, et ce dans tous les domaines de l'ingénierie (voir [Chan and Wu, 2002]). L'objectif du QFD est d'assurer une démarche qualité tout au long du processus de conception, comme le rappelle Yoji Akao, le père de la démarche :

« Le Q.F.D. est une méthode pour développer une qualité de conception visant à satisfaire le client et à traduire les attentes du client en objectifs de conception et en points clés de vérification de la qualité à utiliser durant les phases de production. » ([Akao, 1990])

Le QFD consiste en plusieurs activités supportées par des tables et des matrices. La méthode est décomposée en six étapes dans [Hsiao, 2002] :

1. Identification des clients.
2. Détermination des requis du client.
3. Détermination de l'importance relative des requis.
4. Benchmarking de la concurrence.
5. Traduction des requis du client en requis d'ingénierie mesurables.
6. Mise en place des objectifs de l'ingénierie pour la conception.

Généralement cette méthode s'appuie sur l'utilisation de quatre matrices. Ces matrices sont basées sur le principe du « Quoi-Comment » (cf. [Temponi et al., 1999]). Ces matrices mettent en relation les données produites à une étape du processus avec les décisions devant être prises à l'étape suivante (voir figure I.8.). La première matrice est la matrice de planification qui permet la traduction des attentes du client (Quoi ?) en descriptions techniques (Comment ?), il s'agit de la matrice appelée « *House of Quality* » dans [Hauser and Clausing, 1988]. La seconde matrice met en relation les caractéristiques du produit à concevoir avec la description technique des performances à atteindre. Enfin les caractéristiques des procédés et les requis de production sont

mis en relation avec les caractéristiques d'ingénierie et de marketing dans les troisième et quatrième matrices.

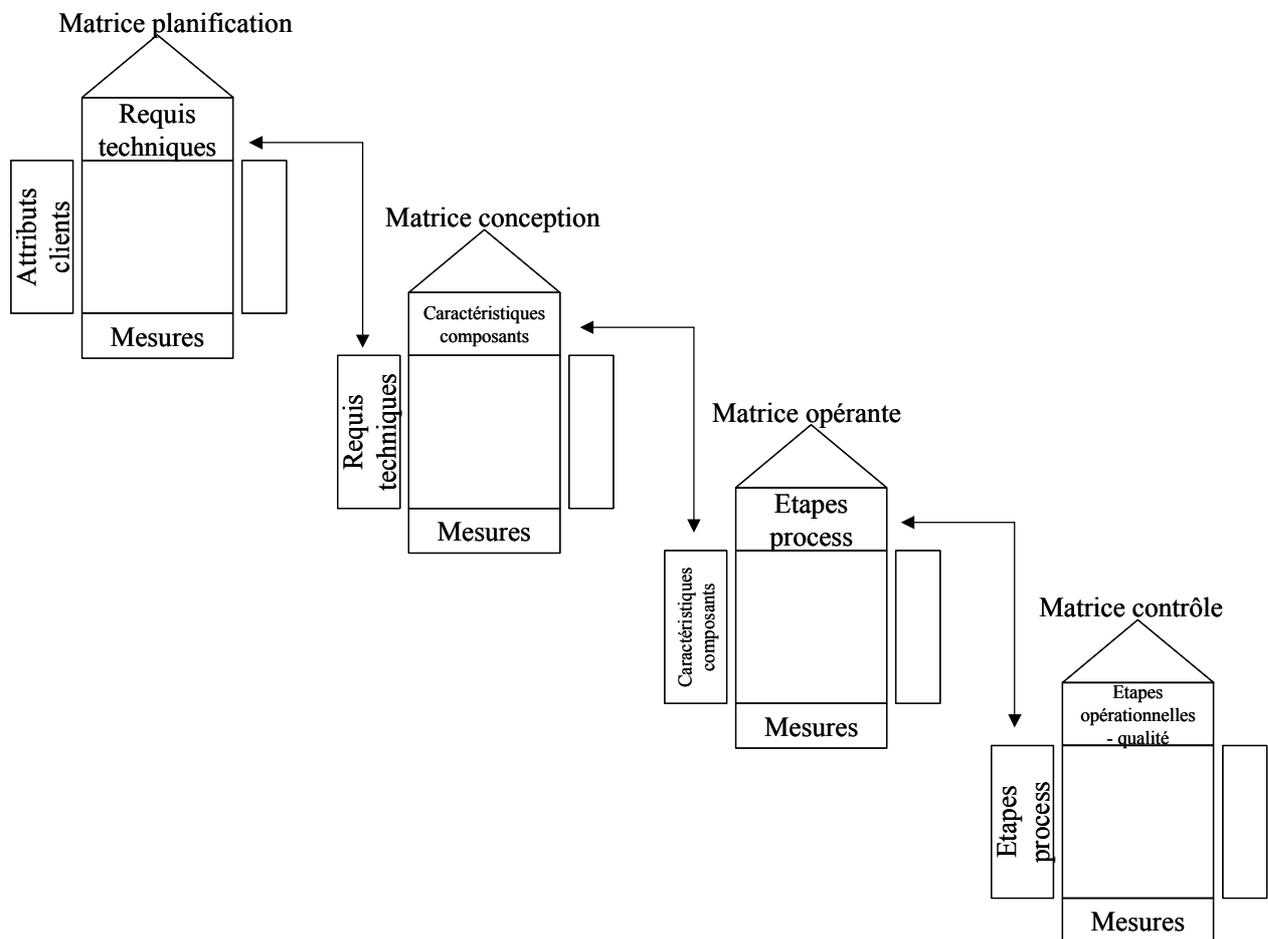


Figure I.8. Le processus de déploiement du QFD, selon [Hauser and Clausing, 1988]

1.2.4.2. Intérêts vis à vis de la résolution de problèmes

La démarche QFD améliore la compréhension des facteurs de succès ainsi que les études de marché orientées sur les attentes du client ([Hutton, 1997]). Elle permet de produire des informations concises et efficaces, en vue des prises de décision jalonnant les processus de conception, production et vente d'un produit ([Schlueter, 2001]). La construction du domaine fonctionnel, si l'on reprend le vocabulaire de Suh, est facilitée par l'approche QFD, qui établit un lien clair et optimal pour le passage du domaine client au domaine fonctionnel, notamment grâce à la matrice « *House Of Quality* ». L'instanciation de cette matrice est une phase de modélisation de problèmes.

En revanche le QFD n'est pas un outil de résolution de problèmes (cf. [Leon-Rovira and Aguayo, 2001]). Le QFD facilite l'identification d'une hiérarchie de problèmes par la hiérarchisation des fonctions à réaliser, en regard des attentes du client. Le domaine de formulation du problème est donc diminué, mais l'identification du cœur du problème et la manière de le résoudre ne sont pas du ressort du QFD.

I.2.5. La TRIZ, théorie de résolution des problèmes inventifs

I.2.5.1. La TRIZ, une approche centrée sur les problèmes

La TRIZ est une théorie de résolution des problèmes inventifs proposée par Altshuller ([Altshuller, 1988]). La TRIZ, acronyme russe pour théorie de résolution des problèmes inventifs, est fondée sur un ensemble de notions essentielles, de méthodes de formulation et de résolution de problèmes, ainsi que sur des méthodes de déblocage de l'inertie psychologique.

Les notions de base de cette théorie sont celles :

- d'idéalité, permettant de définir une solution optimale, souvent utopique, mais qui fournit à la fois une direction d'évolution et un critère d'évaluation des solutions formulées par rapport à cet idéal ;
- d'inertie psychologique : tout concepteur s'appuie sur une certaine expertise acquise au cours de sa formation, de ses expériences, qui lui font décrire et aborder tout nouveau problème au travers d'une vision spécifique. Celle-ci peut être réductrice dans le cadre d'une conception inventive et empêcher le concepteur de formuler des solutions issues de domaines autre que celui de son expertise ;
- des lois d'évolution : tout système technique évolue selon un certain nombre de lois qui permettent de décrire les invariants d'évolution au cours du temps. Ces lois sont basées sur le cycle de vie des systèmes, sur l'analyse de leur potentialité en termes économiques et de brevets. Leur connaissance permet d'anticiper les générations futures d'un produit et d'évaluer le degré d'évolution d'un système ;
- de contradiction : tout problème est issu d'une opposition entre un souhait d'évolution lié à une condition spécifique de l'environnement du système et une loi objective qui empêche la réalisation de cette évolution. Il s'agit d'identifier ces conditions spécifiques et lois objectives afin de pouvoir les contourner au mieux.

La TRIZ aborde la conception avec un point de vue résolution de problème. Elle s'appuie de fait fortement sur les méthodes de re-formulation de problèmes et de résolution par l'analogie. Afin de permettre un tel raisonnement par analogie, Caplat ([Caplat, 2002]) définit qu'il est nécessaire de disposer d'un modèle de formulation des problèmes pour les classer et les comparer à des solutions génériques. Dans la TRIZ, les méthodes de formulation de problèmes s'attachent à identifier systématiquement une contradiction inhérente à toute situation

problématique en analysant les besoins d'évolution ainsi que les lois s'opposant à la réalisation de ces besoins (cf. figure I.9.).

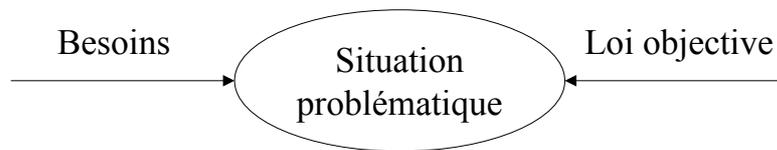


Figure I.9. Une contradiction au cœur de tout problème

1.2.5.2. ARIZ, une méthode de re-formulation de problèmes

La méthode principale de re-formulation et de résolution des problèmes est ARIZ ([Altshuller, 1999]). Cette méthode aborde la résolution de mini problèmes, ces mini-problèmes sont des problèmes fortement contraints, car ils imposent une résolution en apportant le moins de modifications possible. ARIZ se présente sous la forme d'un algorithme (voir figure I.10.) dont l'intérêt est d'identifier de manière systématique :

- l'état que l'on souhaite atteindre, faisant intervenir la notion d'idéalité ;
- la raison pour laquelle cet état nous intéresse, afin de valider le bien-fondé du mini-problème considéré ;
- la loi objective qui nous empêche, dans la situation actuelle, d'atteindre l'état désiré ;
- la raison pour laquelle nous ne pouvons éliminer cet obstacle, qui sont les difficultés de contournement de la loi précédemment identifiée ; ce faisant les caractéristiques du produit à concevoir sont identifiées.

Outre les étapes de re-formulation de problèmes considérées ci-dessus, ARIZ intègre également des étapes d'évaluation et d'exploitation des solutions, d'analyse du processus de conception effectué afin d'améliorer celui-ci et de capitaliser au mieux l'étude réalisée.

L'intérêt d'ARIZ est d'intégrer l'ensemble des outils de la théorie, qui permettent des formulations et résolutions de problèmes à différents niveaux d'abstraction. On peut citer notamment parmi ces outils, les principes d'élimination des contradictions physiques et les standards de résolution. L'objectif commun de ces outils est de se servir de l'abstraction pour établir des analogies avec des solutions ayant résolu des problèmes similaires à celui considéré.

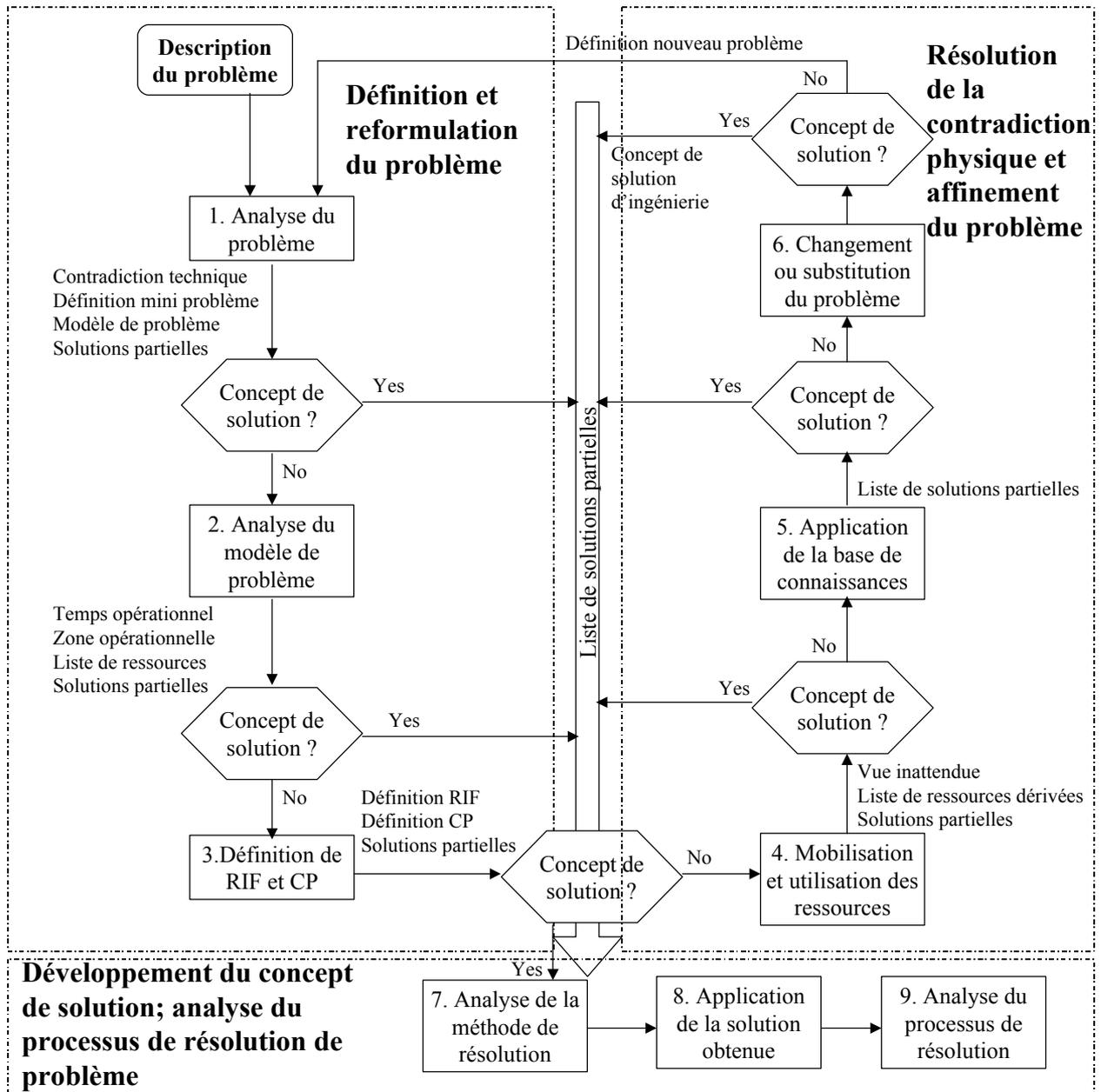


Figure I.10. Processus d'ARIZ, selon [Kucharavy, 2001a]

I.2.5.3. Apports et limites

L'apport premier de la TRIZ, ce en quoi il se distingue des autres approches, est le fait de centrer la processus de conception sur la résolution de problèmes. La TRIZ est ainsi la seule à proposer de définir le problème par une re-formulation visant à réduire le domaine de recherche de la solution. En outre les outils construits sont une passerelle permettant de construire un lien entre domaine fonctionnel et domaine physique (voir [Dubois and Lutz, 2003b]).

Par ailleurs, la vue systémique proposée par la théorie est un complément des plus intéressants à une analyse fonctionnelle. L'apport de cette description morphologique peut troubler mais éclairer la synthèse, voire la genèse, des systèmes. Appuyée par les lois d'évolution des systèmes techniques, elle apporte un outil d'anticipation technologique très intéressant (cf. [Crubleau, 2002]).

Quant à l'algorithme ARIZ, qui permet de clarifier le problème à résoudre et intègre également l'ensemble des notions de la théorie, elle est une méthode puissante mais également délicate à mettre en œuvre, sans une connaissance suffisante de la TRIZ. L'ensemble des compétences à acquérir, en vue d'appliquer avec efficacité la méthode, est spécifique à l'approche de la théorie et a été définie par Khomenko ([Khomenko, 2001]).

Il est donc nécessaire d'identifier les éléments de la théorie, d'accroître le niveau de formalisation de la théorie en construisant une représentation claire et explicite des concepts inhérents. Cette étude, outre le fait de permettre d'assurer l'efficacité de la TRIZ en termes d'utilisabilité et de transfert, permettrait de jeter les bases d'une ontologie de la résolution des problèmes en conception.

1.3. Définition de la problématique

1.3.1. Une vision multi-modulaire de la conception

1.3.1.1. un point de vue particulier de la conception

Comme nous l'avons déjà vu précédemment, il est possible de décrire la conception selon plusieurs points de vue. Nous aimerions ici en proposer une synthèse personnelle, qui permet de décrire les approches selon trois dimensions qui sont sa mise en œuvre, au travers d'outillages et méthodologies, une dimension temporelle, et enfin une dimension segmentaire :

- Un point de vue "axiomes, outils, méthodes" qui permet de classer les notions propres à la conception en fonction de leur mode d'utilisation et de la systématisation de leur utilisation lors de la conception d'un système technique. Une approche de la conception est, de fait, basée sur un des axiomes. L'utilisation d'un outil de résolution de problèmes n'est pas systématique, mais, conditionnée par la typologie du problème à résoudre. Dans ce cas, le degré de connaissances de l'ingénieur vis à vis à la fois du domaine d'application du problème, et de la méthode de conception utilisée, joue un rôle important.
- Un point de vue processus, description temporelle du déroulement de l'étude, est un cadre pour conduire le processus de conception, afin d'en optimiser la gestion. Cette approche a été largement étudiée (Analyse Fonctionnelle, approche systématique de Pahl et Beitz), mais elle ne fournit pas les moyens de répondre aux besoins des concepteurs. Rappelons que le déroulement chronologique d'une étude dépend des acteurs du projet, du temps disponible pour l'étude, du niveau d'inventivité recherché ; De nombreux facteurs rendant le processus de conception spécifique et impropre à la reproduction, sauf à proposer un modèle de processus très générique, mais alors de faible intérêt pratique.
- Un point de vue modules permet de citer de manière générale et exhaustive l'ensemble des étapes à accomplir lors de la conception d'un système technique. Cela sans pour autant indiquer d'ordre chronologique pour l'accomplissement de ces étapes, ni même impliquer une systématisation de réalisation de celles-ci. Une telle description permet d'établir les bases d'une théorie de la conception. Elle fait ressortir la richesse des méthodes par la caractérisation du nombre de modules que les méthodes recouvrent.

Nous proposons dans la suite une description modulaire. Cette description donne une clé de lecture permettant de comparer l'ensemble des méthodes présentées précédemment.

1.3.1.2. Décomposition de la conception en modules

Centrant notre approche de la conception sur la résolution de problèmes (comme le présentent [Austin et al., 2001] et [Kletke et al., 2001]), nous avons proposé une décomposition modulaire intégrant ces deux notions. Dans le tableau I.2., nous listons les modules rencontrés durant le processus de conception, que nous regroupons par phases d'un processus de résolution de problèmes.

Description de la situation initiale	
	Définitions des besoins
	Description de l'environnement
	Définition des spécifications fonctionnelles
Formulation du problème	
	Définition des problèmes
	Classification des problèmes
Résolution des problèmes	
	Définition des ressources
	Outils de résolution
	Lois d'évolution
Evaluation des solutions	
	Cohérence par rapport aux besoins
	Rentabilisation de la solution
	Analyse du processus

Tableau I.2. Décomposition modulaire de la conception

La résolution de problème, en tant que processus, est généralement décomposée en cinq phases ([Gogu, 2001], [Vidal, 1999], [STEM-Net, 1996], [Peltola et al., 2003]) :

1. identification du problème,
2. acquisition des données,
3. analyse des données,
4. génération de solutions,
5. évaluation des solutions.

Cette décomposition ne représente pas un processus linéaire mais l'ensemble des tâches à réaliser. Le processus est itératif, comme le montre bien [Simon, 1987], avec la répétition de plusieurs boucles, en effet, la découverte d'une information conduisant à la re-formulation du problème initial.

Nous avons fait le choix de synthétiser les phases d'identification du problème et d'acquisition des données en une seule et même phase, l'une ne pouvant se réaliser qu'au travers

de l'autre. C'est la phase que nous appelons description de la situation initiale. Cette phase permet de passer d'une idée vague d'un dysfonctionnement à un « problème bien structuré », tel que défini dans [Darlington et al., 1998]. C'est la recherche d'informations qui permet de construire pas à pas un problème bien structuré, c'est à dire un problème conforme aux trois critères suivants :

1. le problème peut être totalement décrit en termes quantitatifs,
2. le but de la tâche peut être décrit par une fonction objective,
3. il existe des algorithmes qui permettent de trouver et de qualifier quantitativement la solution.

Or cette vision, qui décrit le travail du concepteur comme un traitement d'informations, n'est pas habituelle en conception. L'objectif du traitement d'informations est la construction d'un modèle de problème permettant l'utilisation d'algorithmes pour générer la solution. Les outils développés par Altshuller abondent en ce sens, un sens qui a déjà été défini par John Dewey ([Dewey, 1910]) lorsqu'il explicite qu'un problème correctement posé est quasiment résolu.

Cette définition de la situation initiale est composée des étapes classiques de conception que sont la définition des besoins, la description de l'environnement et la définition des fonctions que le système aura à réaliser ([Ullman, 2001]). Sont alors construits entièrement les domaines client et fonctionnel (incluant les contraintes inhérentes à l'environnement d'évolution du système).

Exemple : *problème de transport de laitier, selon le brevet russe 400621 ([Khomenko, 2000a])*

Idée vague de dysfonctionnement : *Durant le transport du laitier de haut fourneau une croûte épaisse se forme. Un tiers du laitier transporté est ainsi perdu. Que faire pour réduire la perte ?*

Problème bien structuré :

La température se forme sous l'action du vent froid. Il a été envisagé de poser un couvercle isolant pour éviter la formation de la croûte mais alors la tâche est rendue difficile : il faut enlever et mettre un couvercle encombrant.

- *Définition des besoins : on souhaite éviter la formation de la croûte.*

- *Description de l'environnement : le transport a lieu du haut fourneau au centre de traitement dans des poches placées sur les plates-formes ferroviaires. Le laitier est en contact avec l'air ambiant. La température de fusion du laitier est de 1000°.*
- *Définition des spécifications fonctionnelles : transporter le laitier en fusion, ne pas modifier le mode de transport, ne pas complexifier le traitement du laitier.*

La seconde phase est celle de formulation du problème, c'est celle qui est la moins abordée dans les approches traditionnelles de la conception. L'analyse des données permet ici de statuer sur le cœur des problèmes à résoudre. La seconde étape de classification est souvent abordée par la hiérarchisation des fonctions à réaliser, sans pour autant analyser la nature des technologies liées à ces fonctions, leur lien de similarité technologique peut apporter une classification différente.

Définition des problèmes : *concevoir un couvercle simple à manipuler, voire auto-destructible, assurer la non formation de la croûte sans couvercle, proposer un traitement de dé-solidification de la croûte.*

Classification des problèmes : *la hiérarchie ne se construit pas ici en termes fonctionnels mais en rapport avec l'idéalité. L'évidence veut que la réalisation de la fonction : « assurer la non formation de la croûte sans couvercle » élimine les deux autres problèmes.*

La résolution des problèmes consiste à la proposition de concepts permettant d'éliminer les problèmes identifiés. Ce processus apporte souvent l'identification de nouvelles informations sur les problèmes, par l'analyse de concepts de solutions partielles. Cette phase peut être réalisée en trois étapes, l'identification des ressources disponibles à la résolution des problèmes, l'exploitation de ces ressources en tant qu'élément de solution et enfin la synthèse ou la modification d'un système selon les lois d'évolution des systèmes techniques. Le mode d'exploitation des ressources peut être outillé afin de faciliter la tâche.

Définition des ressources : *les ressources disponibles à privilégier pour la résolution du problème sont celles déjà présentes, à savoir : le laitier, le contenant, le champ thermique et l'air.*

Ce problème fut résolu par l'utilisation des ressources de laitier et de l'air, avec adjonction d'eau. Le principe est de créer de l'écume de laitier (par l'ajout d'eau) qui a une propriété isolante permettant de résoudre le problème. Un couvercle est ainsi recréé naturellement, couvercle qui disparaît de lui-même lors de déversement.

Enfin, il est nécessaire d'évaluer les concepts générés. D'une part, il faut valider les concepts générés, de par leur cohérence avec les besoins initialement formulés. D'autre part, il est important de rentabiliser la solution par une exploitation optimale du concept généré. Enfin, l'analyse du processus permet de mettre en évidence les erreurs et les pertinences relatives à l'analyse menée et à capitaliser l'expérience.

Nous allons maintenant étudier la pertinence des approches décrites précédemment selon les modules explicités.

I.3.2. Comparaison modulaire des approches

La décomposition proposée va nous permettre de définir l'apport de chaque méthode par rapport aux modules à réaliser au cours de la conception. Nous souhaitons identifier ces apports selon deux niveaux, d'une part la prise en compte de chaque module, deuxièmement le niveau de formalisation de ces méthodes. Cette classification se base sur les travaux de [Cavallucci, 1999; Kurr, 1998; Leon-Rovira and Aguayo, 2001; Mann, 1999; Perrin, 2001; Yang and Zhang, 2000; Yannou and Hajsalem, 2001].

La prise en compte des modules permet de définir le niveau de complétude et les spécificités des méthodes exposées, elle est présentée figure I.11.

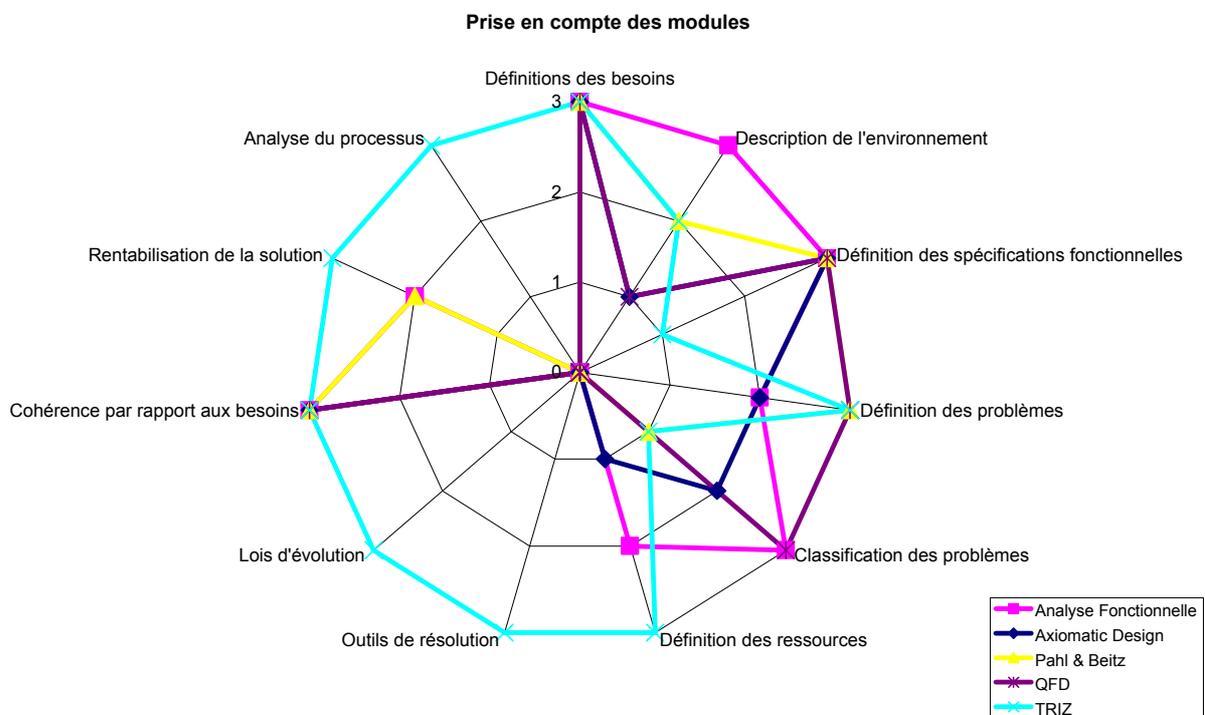


Figure I.11. Prise en compte des modules par les méthodes de conception

On constate sur la figure I.11. qu'aucune méthode ne réalise entièrement l'ensemble des modules de conception. Il serait en revanche possible de construire une méthode complète par intégration (voir [Cavallucci et al., 2000]), l'ensemble des modules étant entièrement pris en compte par l'une ou l'autre des méthodes. Cette approche a été déjà étudiée et deux difficultés majeures en ressortent. D'une part, cela amènerait à proposer une énième méthode parmi la pléthore de l'existant méthodologique, posant le problème de l'intégration d'une méthode dans une entreprise possédant déjà un savoir-faire méthodologique propre. D'autre part, la vision

donnée par la figure I.11. ne fait pas apparaître les différences, souvent notables entre ces méthodes, et les ponts ne sont pas forcément évidents.

Nous préférons avancer qu'il peut être opportun de concevoir, sur la base des méthodes appropriées, un outil pour chaque module à réaliser. Ces outils doivent permettre une réalisation aisée du module concerné, et s'intégrer à toute approche méthodologique, la construction d'un vocabulaire adapté, non ambigu et universel en est la condition. Chaque outil modulaire serait ainsi une partie d'une ontologie de la conception.

La figure fait notamment ressortir la faible prise en compte d'outils de résolution et de lois d'évolution dans les méthodes étudiées, seule la TRIZ les intègre à la conception. C'est donc là une priorité de développement d'outil.

La TRIZ est la seule approche à considérer la nécessité d'outils de résolution de problèmes. Des outils sont proposés par la théorie, toutefois, il semble que leur applicabilité puisse encore être accrue. Comment se fait-il qu'une approche intégrant totalement un module ne le satisfasse pas pleinement ? La réponse est simple, le niveau de formalisation atteint n'est pas suffisant. Nous proposons donc de l'améliorer.

Une seconde vision des méthodes par rapport aux modules est l'étude du degré de formalisation des modules. Nous en présentons le résultat sur la figure I.12.

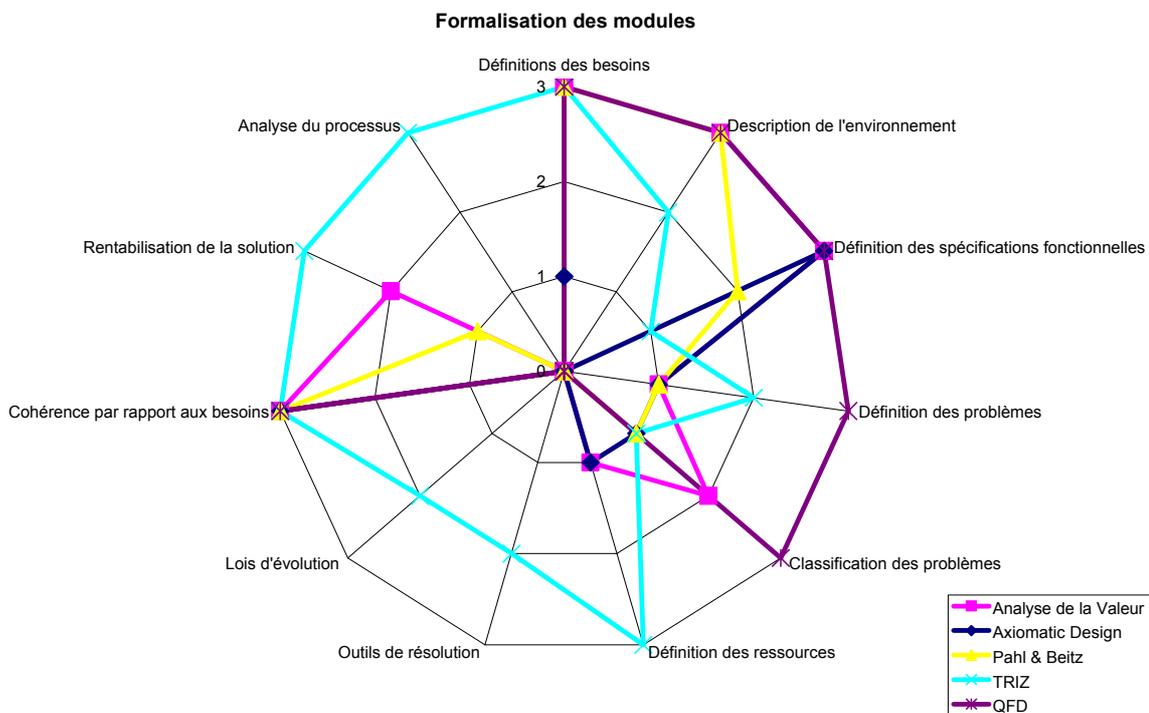


Figure I.12. Formalisation des modules dans les méthodes de conception

Cette description met, là encore, en avant la nécessité de développer des outils proprement formalisés pour les outils de résolution de problèmes et les lois d'évolution. Concernant les lois d'évolution, des travaux existent, notamment ceux de [Crubleau, 2002]. La formalisation d'outils de résolution a été entreprise, par le développement de logiciels, mais pour l'instant rien de satisfaisant n'existe, car les outils formalisés ne sont pas suffisamment génériques.

I.4. Positionnement de la problématique

Des manques ont été identifiés dans l'analyse comparative des approches de la conception : le manque d'outils pour les modules « outils de résolution » et « lois d'évolution ». Seule la TRIZ propose de prendre en compte ces modules. Ces modules semblent pourtant primordiaux dans l'amélioration et l'optimisation de la conception. La TRIZ les intègre pleinement à la démarche de conception, mais leur applicabilité n'est pas optimale car elle un défaut de formalisation existe. Par ailleurs il semble intéressant de pouvoir les intégrer à toute démarche de conception, indépendamment de la méthode suivie.

L'accroissement du formalisme des lois d'évolution et de leur mise en œuvre fait déjà l'objet de travaux. Il semble que des progrès puissent être faits pour la formalisation des modes de résolution de problèmes.

Il est nécessaire de privilégier le développement de la formalisation de la mise en œuvre des outils de résolution de problèmes. Mais pour ce faire, il est nécessaire de formaliser correctement l'étape de modélisation du problème, les outils de résolution de problèmes s'appliquant en fonction de la nature du problème modélisé. La formulation et la résolution du problème sont étroitement liées, résoudre un problème revient à reconnaître un modèle de problème comme concept de solution.

Accroître la mise en œuvre d'outils de résolution passe donc par la construction d'un outil de formalisation de problèmes et la constitution d'une ontologie de formulation de problèmes en conception. Il est d'ailleurs intéressant de noter que la construction du modèle de problème est également un module pour lequel a été identifié un manque de formalisme. Dorénavant nous ne parlerons plus de deux modules distincts, mais de la formulation-résolution des problèmes.

Nous proposons donc de construire une ontologie de modélisation des problèmes en conception pour faciliter la mise en œuvre des outils de résolution de problèmes, sur la base des outils de résolution de la TRIZ.

II. LA TRIZ UNE THEORIE CENTREE SUR LA RESOLUTION DE PROBLEMES

TRIZ, acronyme russe qui signifie Théorie de Résolution des Problèmes Inventifs, est une théorie qui a été élaborée par Genrich S. Altshuller. Cet ingénieur avait été engagé par la marine russe pour étudier les brevets inventifs et accompagner les inventeurs dans leurs démarches de dépôt de brevet ([Cavallucci, 2001]). Durant l'analyse des brevets, il a focalisé son attention sur les processus d'obtention des solutions, indépendamment de la technologie utilisée. Cette démarche particulière a permis d'identifier des concepts génériques ([De Carvalho and Back, 1999]), qui ont constitué les prémices d'une base de connaissances sur l'évolution des systèmes techniques. Il a également constaté, très rapidement, que toutes les bonnes solutions sont obtenues par l'élimination d'une contradiction, comme le rappelle [Arciszewski, 1988]. Dès 1946, Altshuller s'est alors attaché à construire une méthode systématique et générale de résolution de problèmes inventifs. Pour cela, il a puisé dans diverses sources :

- L'analyse des brevets, qui constitue, comme montré précédemment, le cœur de la recherche ;
- La biographie des inventeurs, afin de comprendre l'origine des inventions. Il est intéressant de noter que le domaine d'origine des inventeurs est bien souvent différent de celui de leurs inventions ;
- L'analyse des outils et méthodes occidentales ;
- L'histoire de l'évolution des techniques, afin d'identifier les généralités d'évolution ;
- La psychologie de la créativité, pour comprendre ce qu'est un homme créatif, et comment on peut développer sa pensée créative ;
- La littérature scientifique permettant de lister les effets physiques, chimiques, et autres, qui sont autant de ressources pour résoudre les problèmes de conception ;
- La littérature de science-fiction, enfin, car elle regorge d'idées qui se sont, plus tard, concrétisées, à l'instar des écrits de Jules Verne qui sont un exemple parmi d'autres.

II.1. Le processus de résolution de problèmes de la TRIZ, une construction conjointe du problème et de sa solution

Nous avons commencé à aborder l'importance de la formulation du problème sur l'efficacité du processus de conception. Nous allons maintenant décrire l'approche proposée par la TRIZ pour mener à bien ce processus. Dans un premier temps, nous montrerons l'approche générale de résolution de la TRIZ, qui est proche des méthodes basées sur la réutilisation en Intelligence Artificielle. Puis nous expliciterons l'originalité de la TRIZ, en ce qu'elle génère un processus convergent tout au long du processus de conception. En outre, ce paragraphe s'attachera à montrer que processus de formulation du problème et processus de résolution ne peuvent être disjoints.

II.1.1. Le schéma général de la résolution de problèmes de la TRIZ

L'approche de résolution de problèmes de la TRIZ repose sur l'enchaînement [Altshuller, 1975] :

- description de la situation initiale ;
- identification du problème à résoudre ;
- formulation d'une solution idéale ;
- formulation d'une « solution physique » : solution sous sa forme générale, idée permettant de se rapprocher de la solution idéale ;
- formulation d'une « solution technique » : principe de réalisation de la solution physique, élaboration d'un schéma ;
- formulation d'une solution spécifiée, description complète de la solution, calcul des caractéristiques de la solution.

La TRIZ propose des cadres pour identifier les problèmes à résoudre, la contradiction est le plus répandu de ces cadres. L'intérêt premier des cadres proposés par la théorie est de permettre le recours à des principes généraux de transformation des modèles problèmes. Ces principes sont construits sur la base de lois d'évolution que tout système technique suit au cours de son cycle de vie. Altshuller précise ainsi qu'aucun problème d'ingénierie ne peut être résolu sans le respect des « *directions objectives de l'évolution technologique* » [Altshuller and Shapiro, 1956]. Si de tels principes d'évolution objectifs existent, dans une situation spécifique donnée, permettant de

positionner un objet par rapport à son degré d'évolution, alors les modes d'évolution possibles pour cet objets sont restreints, car contraints par les principes objectifs.

La figure II.1., représentant la démarche générale de résolution des problèmes, spécifiée à la TRIZ illustre ce principe.

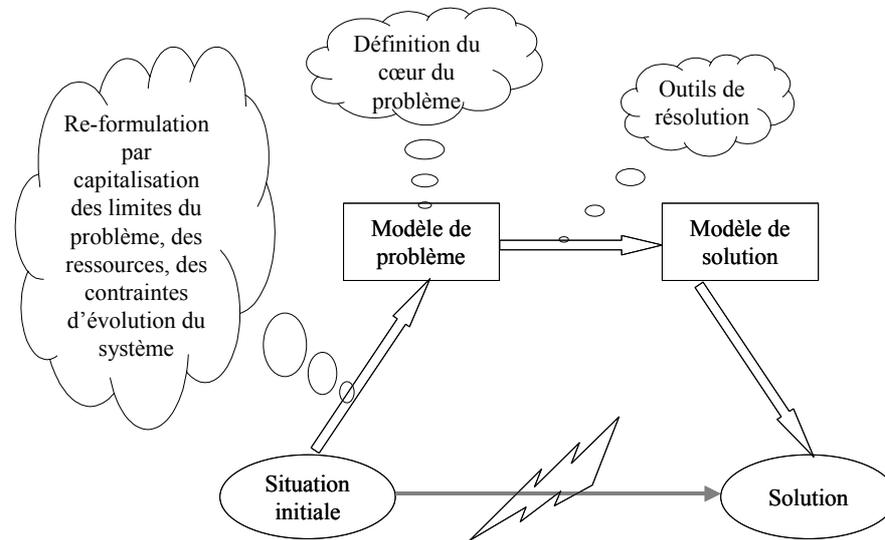


Figure II.1. Approche générale de la TRIZ

Le passage de la situation initiale à l'identification du problème doit permettre :

- de positionner le système étudié dans son évolution,
- de clarifier les possibilités de modification,
- d'identifier les ressources disponibles pouvant être mises en œuvre lors de l'évolution du système,
- de lister les contraintes imposées par l'environnement au système.

Le problème proprement identifié, le mode d'évolution générique du problème se fait en respect des lois objectives d'évolution. La mise en œuvre des outils de la TRIZ font correspondre, à un modèle de problème donné, le modèle de solution adéquat. Suivant la genericité de l'outil employé, i.e. suivant son degré d'abstraction, on aboutit ainsi à la définition de la « solution physique », principe général de la solution, ou de la « solution technique », schéma de réalisation du principe. La synthèse de la « solution spécifiée », intégrant le calcul des caractéristiques principales du système, doit alors permettre de contextualiser le principe de solution en tenant compte des spécificités de la situation initiale [Altshuller, 1975].

Avant de détailler plus avant les outils et les méthodes associées à la TRIZ, nous allons montrer que cette démarche, basée sur la prise en compte de lois objectives d'évolution, conduit à un processus convergent.

II.1.2. Le processus convergent

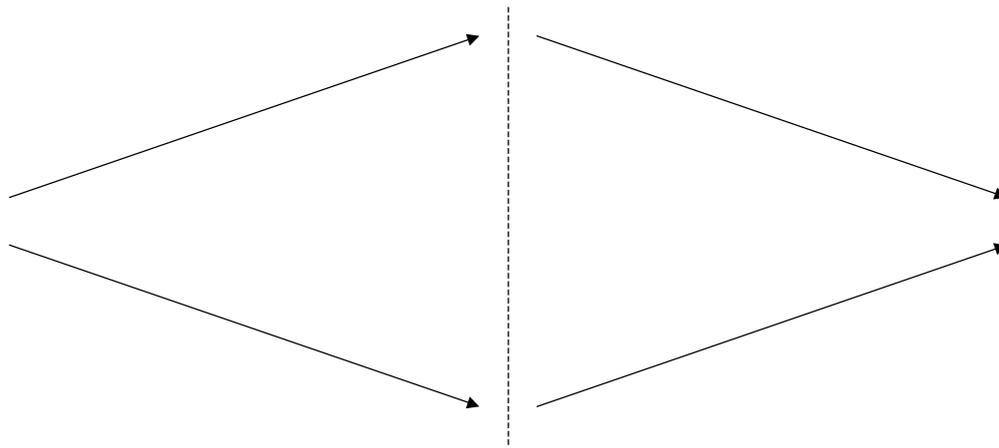
Le processus de conception est jalonné de générations de solutions alternatives qui sont analysées puis sélectionnées. Une modélisation du processus de conception fait généralement ressortir deux logiques d'action de conception ([Perrin, 2001]):

- La divergence, qui a pour effet d'élargir l'espace de recherche de solutions.
- La convergence, qui a pour effet de sélectionner la solution la plus satisfaisante.

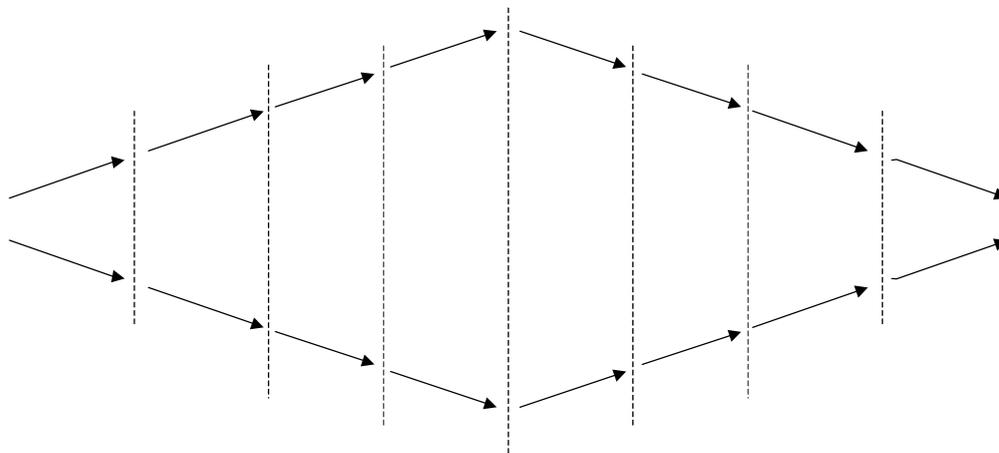
Dans [Liu and Bligh, 2003], Liu caractérise les processus de conception en fonction du nombre et de l'organisation des étapes de convergence et divergence. Trois typologies de conception ont ainsi été identifiées (cf. figure II.2.):

- Un processus simple de divergence, convergence ;
- Un processus multiple de divergence puis un processus multiple de convergence ;
- Un processus multiple de divergence-convergence.

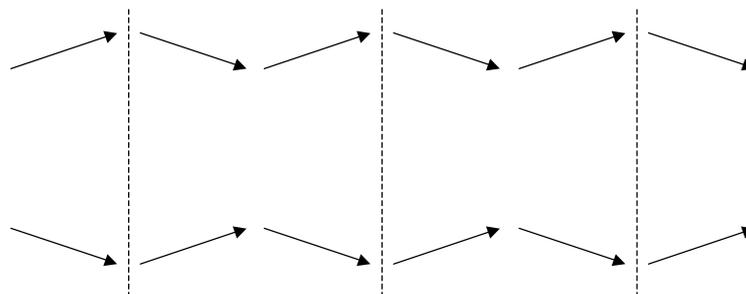
La conception a longtemps été considérée comme un processus ayant pour objectif de développer autant de concepts que possible. La divergence est alors une phase permettant d'accroître le nombre de solutions, en passant d'un niveau abstrait, physique ou technique, à un niveau de solution spécifiée ([Liu and Bligh, 2003]). Pourtant, les concepteurs ne sont pas en mesure de considérer un nombre trop élevé de solutions. Ulrich, dans [Ulrich and Eppinger, 1995], définit ainsi que, si la phase de génération peut amener une centaine de solutions physiques potentielles, seule une vingtaine peut être considérée sérieusement. La convergence doit alors éliminer les solutions les moins appropriées aux données du problème, afin de réduire le nombre de concepts à étudier pour le développement ultérieur.



a. Processus simple de divergence-convergence



b. Processus multiples de divergence, processus multiples de convergence



c. Processus multiples de divergence-convergence

Figure II.2. Les processus de conception, selon [Liu and Bligh, 2003]

En opposition à cette description de la conception, la TRIZ est présentée dans [Gartiser et al., 2002] comme un processus uniquement convergent, se basant sur le postulat, que le nombre de concepts générés n'a pas d'importance, la qualité des idées prévalant. Il est alors nécessaire de quitter le paradigme d'un court temps de recherche de concepts de solution, pour un grand nombre de concepts générés et un temps d'étude des concepts, qui mène à envisager la

conception de produit comme un acte de tri et d'analyse de concepts. La TRIZ préconise un processus convergent, dès la recherche des concepts de solution. Les effets sont alors un allongement de cette phase de recherche, mais un nombre de concepts générés plus petit, induisant un temps d'analyse réduit.

La TRIZ est un processus convergent guidé par deux types de restrictions qui permettent de pointer sur des concepts de solution. Ces restrictions sont, d'une part les lois d'évolution des systèmes techniques et des lois de la physique, et, d'autre part, les conditions spécifiques du problème (cf. figure II.3.).

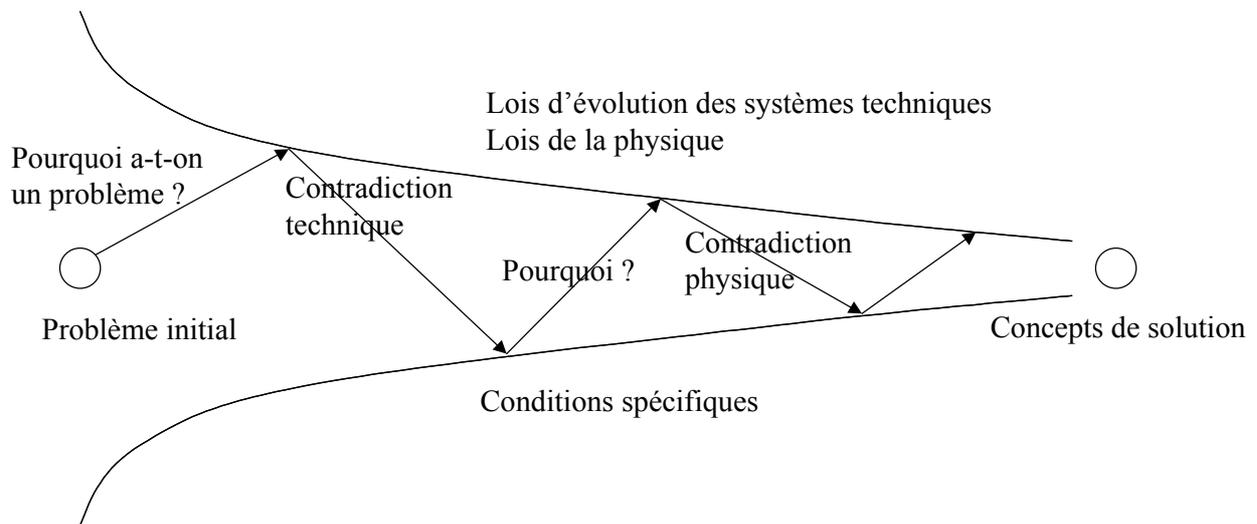


Figure II.3. Le Processus convergent de la TRIZ, selon [Kucharavy, 2001b]

« Le processus de génération de concepts va être un aller-retour entre la définition de la cause d'un problème résultant de conditions de fonctionnement du système traité et la collecte de plus en plus précise des informations permettant sa résolution. » [Gartiser et al., 2002]

L'intérêt du processus de conception convergent est donc, de limiter la phase de sélection des concepts et, de centrer le processus sur l'acquisition des données essentielles au traitement du problème à résoudre. En outre, la réduction de l'aire de recherche, par la prise en compte des conditions spécifiques, permet d'intégrer les contraintes de développement de produit, propres à l'entreprise, et donc d'assurer la viabilité des concepts générés.

II.1.3. Un processus de construction progressif de la solution

L'approche générale présentée sur la figure II.1. montre que les éléments collectés lors de l'identification du problème participent à la spécification de la solution. Ainsi la progression, au cours du processus convergent, permet une synthèse à la fois du modèle de problème et de la solution qui y sera apportée.

Ce phénomène est renforcé par le mode de construction du processus convergent. Les itérations successives, permettant d'abstraire progressivement les données du problème, pour instancier les différents cadres de formulation, permettent une compréhension accrue du problème, ainsi du mode de solution à y apporter. Comme le définit Simon, dans [Simon, 1987], la conception est un processus de génération, de compréhension et de résolution de problèmes et ces trois processus sont finement entrelacés. Il n'est donc pas possible de distinguer formulation du problème et construction de la solution ; formuler un problème, c'est déjà commencer à le résoudre. Cette vision de la conception est partagée par Garro, qui précise dans [Garro, 2000] que le problème est souvent co-défini avec la solution, notamment dans le cadre de conception innovante.

Une analogie avec les mathématiques permet de mieux comprendre ce point, lorsqu'une équation est donnée, par exemple : $x^3 + 14x^2 + 9x = 180$. La recherche des variables solutions passe par la mise en œuvre d'un algorithme de transformation progressif du modèle d'équation jusqu'à l'écriture d'un modèle permettant de reconnaître les variables solutions. Ainsi l'écriture de l'équation : $(x + 5)(x - 3)(x + 12) = 0$, est une re-formulation de l'équation initiale, qui est toujours une formulation de problème mais permettant d'identifier les variables solution.

Les principes, méthodes et outils de la TRIZ sont construits dans le but de guider, à travers un processus systématique, la construction de la solution par des étapes de clarification et de re-formulation du problème. D'une part, les concepts d'idéalité et de lois objectives d'évolution, qui vont être développés dans le prochain paragraphe, orientent la construction de la solution, assurant ainsi la convergence du processus. D'autre part, le mode de construction des modèles de formulation de problème, notamment la formulation de contradictions, qui intègrent le problème à résoudre et les causes d'apparition du problème collectent à la fois les données du problème à résoudre, mais également les caractéristiques que la future solution doit posséder.

II.2. Des concepts de base

La TRIZ repose sur des concepts de base qui sont à l'origine de tout processus de résolution de problèmes, leur formulation étant adaptée aux spécificités de la conception. Nous allons présenter ici l'ensemble de ces notions de base qui sont au cœur de la théorie.

II.2.1. Le Résultat Idéal Final

I.1.3.1. La notion d'idéalité

L'idéalisation est une approche répandue en sciences ([Savransky, 2000]). Savransky la définit comme un acte mental de création d'objets abstraits, ne pouvant exister en réalité et ne pouvant être obtenus par quelque expérimentation. Ces objets idéaux représentent une limite pour les objets réels. Les objets idéaux (le point ou la ligne en géométrie, les gaz idéaux en physique) jouent un rôle important pour le développement des théories axiomatiques et l'analyse des objets réels. L'idéalité peut alors se définir comme une abstraction permettant des représentations de la réalité, utiles à l'études de divers phénomènes.

La TRIZ inclut la notion d'idéalité qui peut se décliner en système idéal, processus idéal, ressource idéale, solution idéale, ... suivant le point de vue et la phase de conception. On peut ainsi parler de manière plus générale d'élément idéal. Un élément idéal se reconnaît en ce qu'il n'a aucun poids, n'occupe aucun espace, ne consomme aucune énergie, ne coûte rien, ne nécessite pas de maintenance, ... mais accomplit le travail, la fonction requise.

Un critère d'idéalité, qui sert à évaluer le degré d'idéalité d'un concept, est proposé et

défini par l'équation :
$$I = \frac{\sum F_u}{\sum F_n + \sum F_c}$$
, où I désigne le critère d'idéalité, F_u les fonctions utiles – fonctions agissant sur l'environnement de manière utile, F_n les fonctions néfastes – fonctions nuisant à l'environnement, et F_c les fonctions coûts. Ainsi accroître l'idéalité revient à maximiser les fonctions rendues par le système et à minimiser les fonctions néfastes et les fonctions coûts.

I.1.3.2. Le Résultat Idéal Final

« Le RIF vous évite de dire une sottise. Dans l'activité innovante, le silence est forcément apprécié, mais quand on dit quelque chose d'intelligent c'est de l'or... » [Altshuller, 1975]

Le Résultat Idéal Final (RIF) est la métaphore de la solution idéale pour un problème donné, sous des conditions données. Le RIF est formulé selon une méthode très simple :

« L'un des éléments impliqués dans la contradiction supprime de lui-même l'effet néfaste, tout en préservant sa capacité à réaliser son action de base. » [Altshuller, 1988]

Pour Altshuller (dans [Altshuller, 1975]) le RIF reflète la voie d'évolution de la technique, le fait que chaque machine soit plus performante que la précédente. Il permet de séparer les solutions appartenant à des niveaux d'inventivité faible des solutions appartenant aux niveaux supérieurs. Etant donné que les premières sont nettement plus nombreuses que les secondes, le RIF réduit considérablement l'aire de recherche de la solution.

La formulation du RIF doit donc faire partie de tout processus de conception. Le RIF donne à la fois un critère d'évaluation des concepts générés et la direction d'évolution générale du système considéré, il participe, à ce titre, fortement à la convergence du processus.

II.2.2. Les lois d'évolutions des systèmes techniques ([Altshuller, 1988; Salamatov, 1996])

II.2.2.1. Introduction

« L'évolution de la technique, en tant qu'élément de l'évolution progressive de la civilisation, est un processus historique autonome soumis à des lois, qui non seulement ne sont pas dépendantes de la volonté de l'homme, de sa conscience et de ses intentions, mais bien au contraire les déterminent. » [Salamatov, 1996]

Ainsi l'homme serait soumis au respect d'invariants d'évolution, qui réduisent ses choix de conception. Pourtant un concepteur ne ressent pas de limites à sa créativité, et il n'est effectivement pas limité. L'homme peut concevoir un système ne respectant aucune loi d'évolution, ce qu'il fait d'ailleurs souvent. Nous appelons cela un processus d'essais-erreurs. Car l'homme peut ignorer les lois d'évolution, mais, ce faisant, le système conçu ne pourra remplir la fonction pour laquelle il est destiné. Cette pensée peut induire une certaine crainte technique chez l'homme prisonnier de la machine (voir [Heidegger, 1953]), crainte stigmatisée par Ellul :

« Ce n'est pas la présence de l'homme qui empêche la Technique de se constituer en système : l'homme qui agit et pense aujourd'hui ne se situe pas en sujet indépendant par rapport à une technique objet mais il est dans le système technique, il est lui-même modifié par le facteur technique. » [Ellul, 1977]

André Leroi-Gourhan ([Leroi-Gourhan, 1971]) voit dans l'activité humaine deux ordres de phénomènes de natures distinctes :

- des phénomènes de *tendances* qui tiennent à la nature même de l'évolution ;
- des phénomènes de *faits* qui sont indissolublement liés au milieu dans lequel ils se produisent.

Il y aurait donc une tendance technique universelle et rayonnante, que les différents groupes ne font qu'interpréter selon leurs propres besoins. La tendance mène donc toujours à une expression locale de la technique. Si la tendance a un caractère inévitable prévisible et rectiligne ; le fait, quant à lui, est imprévisible et particulier ; c'est tout autant la rencontre de la tendance et des mille coïncidences du milieu, que l'emprunt pur et simple à un autre peuple. Ainsi l'apparition de l'outil à trancher respecte la tendance universelle, en apparaissant quasi-

simultanément à travers le monde, sa réalisation étant spécifique aux matériaux, us et environnements locaux.

Genrich Altshuller ([Altshuller, 1988]) s'est intéressé à formaliser les constantes d'évolution suivies par les systèmes techniques. La connaissance de ces constantes permet de synthétiser des systèmes respectant un sens cohérent d'évolution ; par ailleurs cette connaissance permet d'anticiper les évolutions et de prédire l'apparition des problèmes (voir [Salamatov, 1999]). Altshuller a formé un corpus de lois décrites ci-après, classifiées selon trois groupes :

1. Les lois statiques : elles définissent la synthèse de tout système technique opérationnel.
2. Les lois cinématiques : elles définissent le mode d'évolution des systèmes, indépendamment des facteurs physiques et techniques qui réalisent cette évolution ([Crubleau, 2002]).
3. Les lois dynamiques : elles traduisent l'évolution des systèmes techniques modernes sous l'effet de facteurs physiques et techniques.

II.2.2.2. Les lois statiques

« Tout système technique résulte de la synthèse de parties séparées en un tout. Mais toute synthèse des parties est loin de pouvoir former un système technique apte à fonctionner. Il existe au moins trois lois dont les règles doivent être respectées pour que le système technique soit opérationnel. » [Altshuller, 1977]

II.2.2.2.1. Loi d'intégralité du système technique

Une condition indispensable pour qu'un système technique fonctionne est une aptitude minimale des parties principales du système technique à fonctionner. Chaque système technique comporte quatre parties principales (voir figure II.4.), à savoir :

- un moteur qui transforme l'énergie provenant de l'environnement en énergie utile au système ;
- une transmission qui transmet l'énergie du moteur vers l'organe de travail ;
- un outil qui réalise physiquement la fonction du système ;

- un organe de contrôle qui permet l'adaptation fonctionnelle du système.

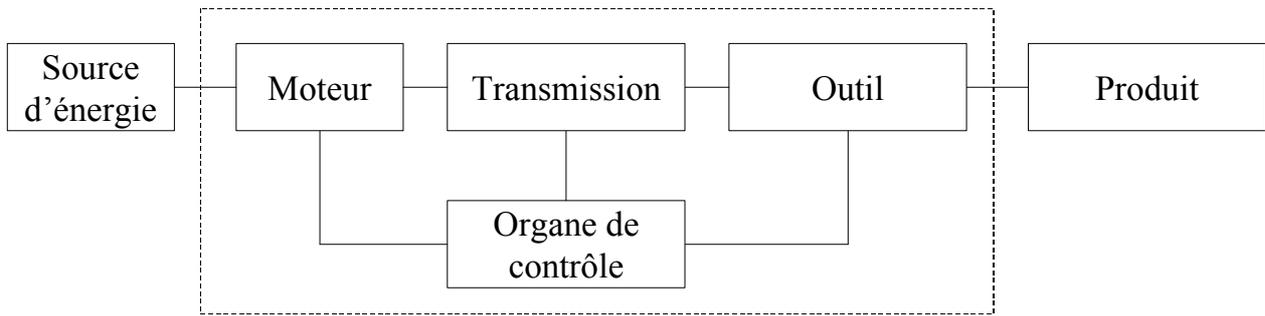


Figure II.4. Intégralité d'un système technique, selon [Salamatov, 1996]

Une condition nécessaire au bon fonctionnement d'un système est également la contrôlabilité d'au moins l'une des parties le composant.

II.2.2.2.2. Loi de conductibilité énergétique des parties

Le bon fonctionnement d'un système technique nécessite le libre passage de l'énergie à travers toutes ses parties. C'est une condition indispensable.

Tout système technique est à la fois conducteur et convertisseur d'énergie. Si l'énergie ne traverse pas l'ensemble du système, autrement dit, si elle reste bloquée quelque part, une des parties manquera d'énergie et, par conséquent, ne fonctionnera pas.

Pour qu'une partie d'un système soit contrôlable, il est indispensable d'assurer un libre passage de l'énergie entre ladite partie et son organe de commande.

II.2.2.2.3. Loi de concordance des rythmes

La mise en concordance (ou la discordance intentionnelle) de la fréquence, des oscillations (ou de la périodicité de fonctionnement) de toutes les parties d'un système technique est une autre condition indispensable à son évolution.

Les systèmes fonctionnant correctement sont ceux dans lesquels un type de vibration est sélectionné de manière à ce que leurs parties ne se contrarient pas et remplissent au mieux la fonction utile. Ils sont, par voie de conséquence, opérationnels.

II.2.2.3. Les lois cinématiques

« La cinématique comprend des lois qui définissent l'évolution des systèmes techniques sans tenir compte des facteurs techniques et physiques concrets déterminant cette évolution. »
[Altshuller, 1977]

II.2.2.3.1. Loi d'augmentation du degré d'idéalité

L'évolution de tout système technique tend vers le niveau le plus élevé d'idéalité.

Un système considéré idéal est celui dont le poids, le volume et la surface cherchent à atteindre zéro et dont la capacité de travail reste toujours la même. Autrement dit, c'est un système qui n'existe pas mais dont les fonctions sont conservées et remplies.

Nous rappelons le critère permet de mesurer cette idéalité :
$$I = \frac{\sum F_u}{\sum F_n + \sum F_c}$$

Cette loi est très générique, deux voies de réalisation peuvent être identifiées :

- L'accroissement de l'idéalité par l'augmentation des fonctionnalités et la réduction des défauts, par intégration de substances voire de sous-systèmes.
- L'accroissement de l'idéalité par la dynamisation des systèmes, afin d'augmenter l'adaptation du système à son environnement, en passant d'un système monobloc rigide à des systèmes souples puis à des champs.

II.2.2.3.2. Loi de développement égal des parties

Les parties du système se développent inégalement : plus le système est complexe, plus le développement de ses parties est inégal. Ce qui est générateur de l'apparition de contradictions physiques et techniques et, par conséquent, de problèmes d'invention.

Il est nécessaire de faire évoluer les systèmes techniques en respectant une homogénéisation du développement de chaque partie du système.

II.2.2.3.3. Loi de transition vers un super-système

L'évolution d'un système qui a atteint sa limite peut se perpétuer au niveau d'un super-système.

Voici une des voies de cette transition : *des systèmes techniques se réunissent et forment des bi-systèmes et des poly-systèmes.*

La réunion des systèmes en un super-système comporte quelques avantages :

- Certaines fonctions des systèmes se transmettent à un super-système .
- Certains sous-systèmes sortent du système technique et, après s'être réunis en un seul système, deviennent une partie du super-système.
- Les systèmes réunis en un super-système acquièrent de nouvelles fonctions et propriétés.

II.2.2.4. Les lois dynamiques

« La dynamique comprend des lois qui traduisent l'évolution des systèmes techniques modernes sous l'effet de facteurs techniques et physiques concrets. Les lois statiques et cinématiques sont universelles ; en d'autres termes, elles sont vraies de tout temps, non seulement pour les systèmes techniques, mais aussi pour les systèmes en général (biologiques, sociaux, etc.). La dynamique exprime des tendances principales d'évolution des systèmes techniques uniquement à notre époque. » [Altshuller, 1977]

II.2.2.4.1. Loi de transition du macro-niveau au micro-niveau

Les organes de travail évoluent d'abord au macro-niveau – au niveau des systèmes et des éléments du système, puis au micro-niveau – au niveau de la substance.

Ainsi, on va passer d'un travail réalisé par des roues, des arbres et des roues dentées, à un travail réalisé par des molécules, des atomes, des ions, des électrons, etc. Ces derniers sont alors plus faciles à contrôler par des champs, à l'aide des effets physiques et chimiques.

Il existe trois voies de transition du macro-niveau au micro-niveau :

- L'augmentation du degré de segmentation de la substance et la réunion des parties segmentées en un nouveau système.
- L'augmentation du degré de segmentation du "mélange" de substance et de vide (transition vers les matériaux capillaro-poreux).
- La substitution, dans un système, d'un champ à une partie matérielle.

II.2.2.4.2. Loi d'augmentation de la « vépolisation³ »

A ce niveau, nous associons « augmentation de la vépolisation » à l'augmentation du niveau de contrôle des systèmes.

Le développement des systèmes techniques tend vers un niveau plus élevé de « vépolisation » : les systèmes non-vépolisés, assimilés à des systèmes non contrôlés ou à des relations entre constituants mal construites, cherchent à devenir vépolisés. Par contre, à l'intérieur des systèmes vépolisés, le développement s'effectue par la multiplication des liens entre éléments, ainsi que par l'augmentation de la sensibilité et du nombre d'éléments.

II.2.2.5. Conclusion

La vision systémique, proposée par ces lois, est un apport inestimable à toute étude prospective, et même à la conception de tout produit technique. La description fonctionnelle, telle qu'elle est proposée, apporte une vision nouvelle sur la synthèse des systèmes.

En reprenant la classification de [De Araujo, 1996], on peut considérer que les lois énoncées représentent un ensemble structuré de connaissances. Il est nécessaire de construire des outils sur cette base. C'est ce qui est mis en avant par [Zakharov, 1993], et ce à quoi Crubleau propose une solution dans [Cruleau, 2002], voire encore Kozyreva dans ses travaux ([Kozyreva, 2002]) destinés plus spécifiquement à l'enseignement.

³ Les Vépoles sont des modèles permettant de représenter les systèmes selon des assemblages de substances et de champs. Nous détaillons cet outil au paragraphe II.3.2.

II.2.3. Tout problème est une contradiction

Comment passer d'une formulation d'un besoin fonctionnel à la reconnaissance de la structure physique permettant sa réalisation ? Dans [Altshuller, 1988], il est proposé de décrire un tel problème (je veux réaliser une fonction mais je ne sais pas comment) sous la forme d'une contradiction, cause commune de tout problème créatif. **La TRIZ propose ainsi, plutôt que de tenter de résoudre directement un problème, de formuler une contradiction, et de l'éliminer en suivant les tendances génériques d'évolution des systèmes** ([Killander and Sushkov, 1995]). Linde et al. reprendront ce principe pour développer WOIS, voir [Linde and Herr, 2002]. WOIS est basé sur l'identification de contradictions à résoudre pour suivre les tendances générales d'évolution. La formulation de contradictions résulte d'une analyse exhaustive des objectifs d'évolution et des composants du système à faire évoluer.

Altshuller a mis en évidence l'existence de trois niveaux de formalisation des contradictions : administratif, technique et physique. Aujourd'hui les experts s'accordent à dire que cette dénomination n'est pas la plus appropriée ([Savransky, 2000]), surtout si l'on cherche à établir cette classification pour des domaines autres que l'ingénierie, comme les travaux du Projet Jonathan Livingstone (voir [Khomenko and Sokol, 2000]). En revanche, l'existence de différents niveaux de formalisation est utile, car elle est représentative des différents niveaux de compréhension du problème.

II.2.3.1. La contradiction administrative

« *Je sais quoi, mais je ne sais pas comment !* » [Cavallucci, 2001]

Un problème formulé ainsi fait ressortir le besoin de réaliser une fonction, d'éviter un phénomène indésirable, mais n'offre aucune voie de résolution. Par exemple, les besoins d'augmenter la productivité ou de réduire les coûts sont des contradictions administratives. Un tel problème soulève une situation inventive. La contradiction administrative est une formulation provisoire qui n'a pas de valeur heuristique, et n'offre aucune direction de recherche de la solution.

II.2.3.2. La contradiction technique

« *Je sais comment, mais à cause de ça, ça devient pire !* » [Cavallucci, 2001]

Dans le cas de la contradiction technique, une action est à la fois utile et néfaste ; l'intégration ou l'amplification de l'action utile, ou la suppression de l'action néfaste, entraîne la détérioration d'un sous-système, voire du système tout entier. La contradiction technique

représente un conflit entre deux sous-systèmes. Par exemple, l'augmentation de la profondeur de pénétration des ions dans un semi-conducteur entraîne l'accroissement de la puissance électrique nécessaire (exemple issu de [Savransky, 2000]). De telles contradictions apparaissent si :

- La création, ou l'intensification d'une fonction utile dans un sous-système crée une nouvelle fonction néfaste, ou intensifie une fonction néfaste, existante dans un autre sous-système.
- L'élimination ou la réduction d'une fonction néfaste dans un sous-système détériore une fonction utile dans un autre sous-système.
- L'intensification d'une fonction utile, ou la réduction d'une fonction néfaste, dans un sous-système cause une complexification inacceptable d'autres sous-systèmes, voire du système entier.

II.2.3.3. La contradiction physique

« *Je sais quoi et comment, mais je ne sais pas avec quel moyen !* » [Cavallucci, 2001]

Un sous-système donné doit avoir la propriété A pour réaliser une fonction nécessaire, et la propriété non-A pour satisfaire les données du problème. Lorsque dans un système on améliore une pièce (ou un paramètre), une autre pièce (ou paramètre) s'en trouve détériorée. Bien souvent cette contradiction technique est cachée ou mal formulée. Par exemple, il est souhaité d'avoir une faible constante diélectrique de l'isolant des puces de semi-conducteurs afin de réduire les capacités parasites, mais il est préférable d'avoir une grande constante diélectrique de l'isolant pour stocker l'information. Une contradiction physique est systématiquement cachée derrière une contradiction technique, il s'agit alors de trouver l'élément portant la contradiction.

II.2.3.4. Conclusion

Les contradictions sont des modèles de représentation de problème qui apportent une gradation de formalisation. Contradictions techniques et physiques sont des modèles permettant des résolutions heuristiques ; la difficulté est de formuler correctement celles-ci. Il est ainsi nécessaire de proposer un outil de formulation de contradiction.

En outre, la contradiction permet une compréhension très explicite du problème, car elle s'appuie sur un raisonnement dialectique. Chosson définit d'ailleurs que pour dégager les problèmes, « *il convient de mettre en évidence les contradictions qui sont au cœur des situations, car c'est en les cernant que l'on parvient à dégager les problèmes.* » [Chosson, 1975].

La contradiction revêt donc une importance particulière dans la formulation de problèmes, **et l'attention doit être portée sur l'identification de la contradiction, inhérente à tout problème de conception.**

II.2.4. Les ressources

Les ressources jouent un rôle important dans la résolution de problèmes, de manière à se rapprocher du RIF. Toute technique, ou produit, n'ayant pas atteint l'idéalité, a encore des ressources de développement disponibles. Tout élément est une partie d'un super-système, et une partie de la nature. En tant que tel un élément existe dans l'espace et dans le temps. Il est composé et/ou utilise des substances et des champs, et réalise des fonctions. Les ressources peuvent donc être classifiées selon qu'elles sont ([Savransky, 2000]):

- Ressources de temps : intervalles de temps avant, pendant et après les réalisations des fonctions.
- Ressources d'espace : position dans l'espace, emplacement dans un sous-système ou dans un super-système, zone fonctionnelle.
- Ressources de l'environnement : toute substance ou tout champ existant dans la nature.
- Ressources de super-systèmes : systèmes adjacents, faisant parties d'un super-système commun.
- Ressources du système : toute substance, tout champ appartenant au système.

Une fois un problème correctement formulé, il est nécessaire d'identifier la ressource qui permettra de résoudre le problème. Le processus de sélection de la ressource doit se faire en considérant la disponibilité, c'est-à-dire la quantité, l'utilisabilité, le bénéfice apporté, la proximité du système et enfin, le coût d'introduction. La figure II.5., d'après [Cavallucci, 2001], synthétise la caractérisation des ressources.

La considération des ressources permet de voir à l'extérieur de la boîte noire une zone et un temps opératoires d'apparition du problème, ce qui est important en phase de conception, quand l'esprit est focalisé sur un système particulier. Ainsi l'utilisation de ressources telles que le vide, les déchets ou les substances et champs disponibles et économiques, permettent la réalisation de systèmes performants. La plupart du temps, les ressources existent dans le système ou dans son super-système, mais elles peuvent être cachées ou difficiles à identifier.

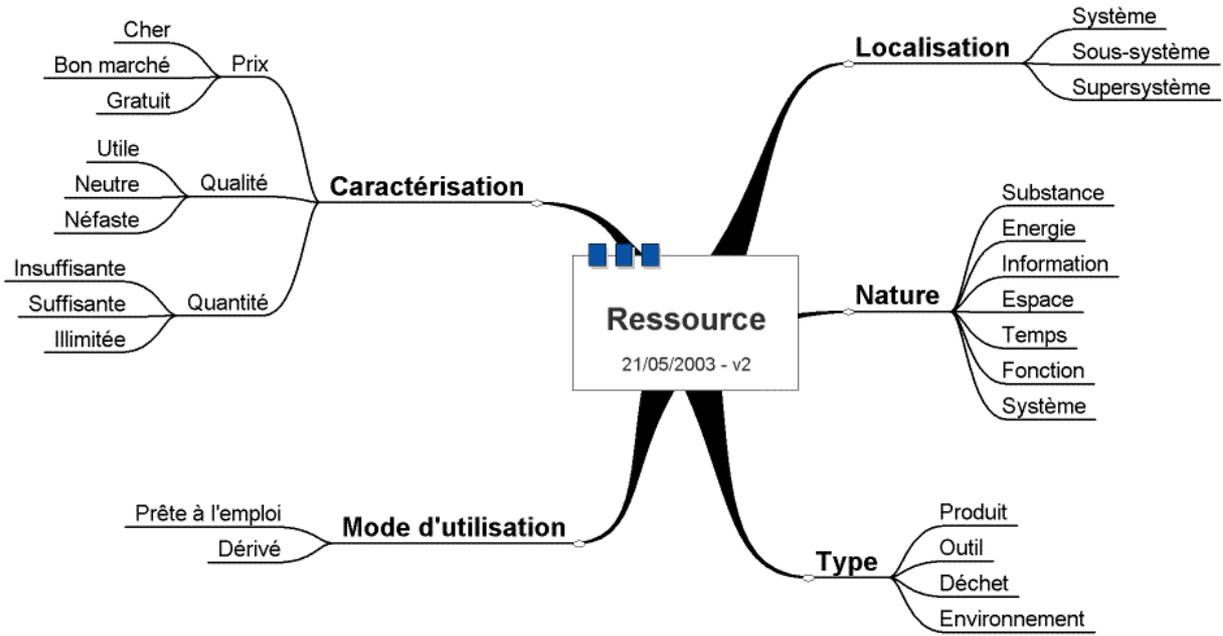


Figure II.5. Caractérisation des ressources, selon [Cavallucci, 2001]

II.2.5. L'inertie psychologique

La difficulté de résolution d'un problème dépend des capacités de la personne à qui le problème est soumis. Une panne de voiture peut s'avérer être un problème complexe pour un scientifique, mais une opération routinière pour un garagiste. A l'inverse, un scientifique peut résoudre des problèmes de conception de voiture qui dépassent l'entendement du garagiste. Ainsi, en fonction de la personne à qui elle se présente, une situation peut s'avérer problématique ou non.

La perception qu'un individu a d'une situation dépend d'un ensemble de facteurs influents (cf. [Chosson, 1975]), que sont : les expériences passées, les connaissances acquises ou encore les habitudes de raisonnement. Tous ces facteurs influent sur la représentation qu'un individu se fait d'une situation et orientent la manière de traiter cette situation. Si cette dernière est problématique, le réflexe d'y trouver une solution dans les connaissances les mieux maîtrisées est naturel et peut se révéler efficace pour des problèmes d'optimisation, mais il peut empêcher la résolution de la situation, notamment lors de conception inventive. Ce phénomène est identifié sous le terme d'inertie psychologique.

L'inertie psychologique est le principal frein à la créativité des individus, avec des éléments déclencheurs tels que :

- Les habitudes.
- Les compétences trop pointues dans un domaine particulier.
- Les inerties générées par le jargon du spécialiste.

Quelques règles permettent de ne pas subir l'inertie psychologique, telles que :

- Ne jamais être persuadé que la solution réside dans son propre domaine de compétence.
- Favoriser la pluridisciplinarité.
- Identifier les termes ou expressions porteurs d'inertie psychologique et les remplacer par d'autres plus neutres.
- Respecter toutes les idées, même les plus farfelues.

Il est nécessaire d'être conscient du phénomène d'inertie psychologique et de faire en sorte de ne pas y être soumis. Pour cela, l'abstraction est un moyen efficace. Le fait de construire la description d'un objet par des paramètres génériques permet de construire des analogies avec des

concepts issus de domaines variés. Par ailleurs la TRIZ comporte un certain nombre d'outils pour dépasser cette inertie, par exemple une méthode basée sur la synectique de Gordon, ou l'utilisation de la génération de situations extrêmes par intensification des opérateurs d'espace, de temps et des ressources disponibles ; ces méthodes sont définies dans [Altshuller, 1988].

II.3. Des méthodes de formulation et de positionnement de problèmes

Nous allons présenter ici les méthodes proposées par la TRIZ, à savoir l'approche multi-écrans, l'algorithme ARIZ de formulation de problèmes, et enfin, un outil particulier de formulation et résolution de problèmes : la modélisation Vépole et les standards de résolution.

II.3.1. L'approche multi-écrans, structure de l'esprit fort⁴

« Les inventeurs peu doués examinent des variantes et des petites modifications des objets pendant longtemps. Ces variantes (même la quinzième version) ne diffèrent que peu de l'original. [...] Un inventeur efficace sélectionne les variantes plus audacieusement. Les images changent rapidement sur son écran mental, et des représentations non usuelles y sont représentées. » [Altshuller, 1988]

Altshuller définit l'esprit fort comme l'art de concevoir des images non usuelles ; pour cela, il décrit l'aptitude à remplir, simultanément, plusieurs écrans relatifs à différents niveaux systémiques d'un produit, échelonnés dans le temps.

Dans un premier temps, il est nécessaire de savoir segmenter le monde en niveaux systémiques : le premier indicateur de l'esprit fort est l'aptitude à passer d'un système au super-système et aux sous-systèmes. Ainsi, lorsque l'on considère un arbre, il faut avoir à l'esprit la forêt (le super-système) et les composantes de l'arbre : les racines, le tronc, les feuilles... (sous-systèmes).

Par ailleurs, pour chacun de ces niveaux systémiques, il faut considérer les lignes d'évolution que sont le passé, le présent et le futur de ces systèmes (cf. figure II.6.).

Ces deux lignes peuvent être répétées autant de fois que nécessaire. Le super-système de la forêt est la biosphère, la feuille est composée de cellules. Plusieurs écrans existent à gauche du système (le passé récent, le passé moyen, le passé lointain) ; il en est de même à droite (le futur proche, le futur moyen, le futur lointain). Altshuller reconnaît la complexité de cette approche multi-écrans, mais il insiste sur la complexité du monde. Si nous souhaitons comprendre et transformer le monde dans lequel nous vivons, notre pensée doit en représenter la complexité.

⁴ Traduction issue de [Kucharavy, 2002] Kucharavy, D. (2002) *Russian-English-French OTSM-TRIZ Dictionary of Conformities*, OTSM-TRIZ Technologies Center, Minsk, Belarus, 1998-2002., nous préférons le terme « Advanced Thinking » de l'OTSM-TRIZ au terme initial d'Altshuller : « Talented Thinking ».

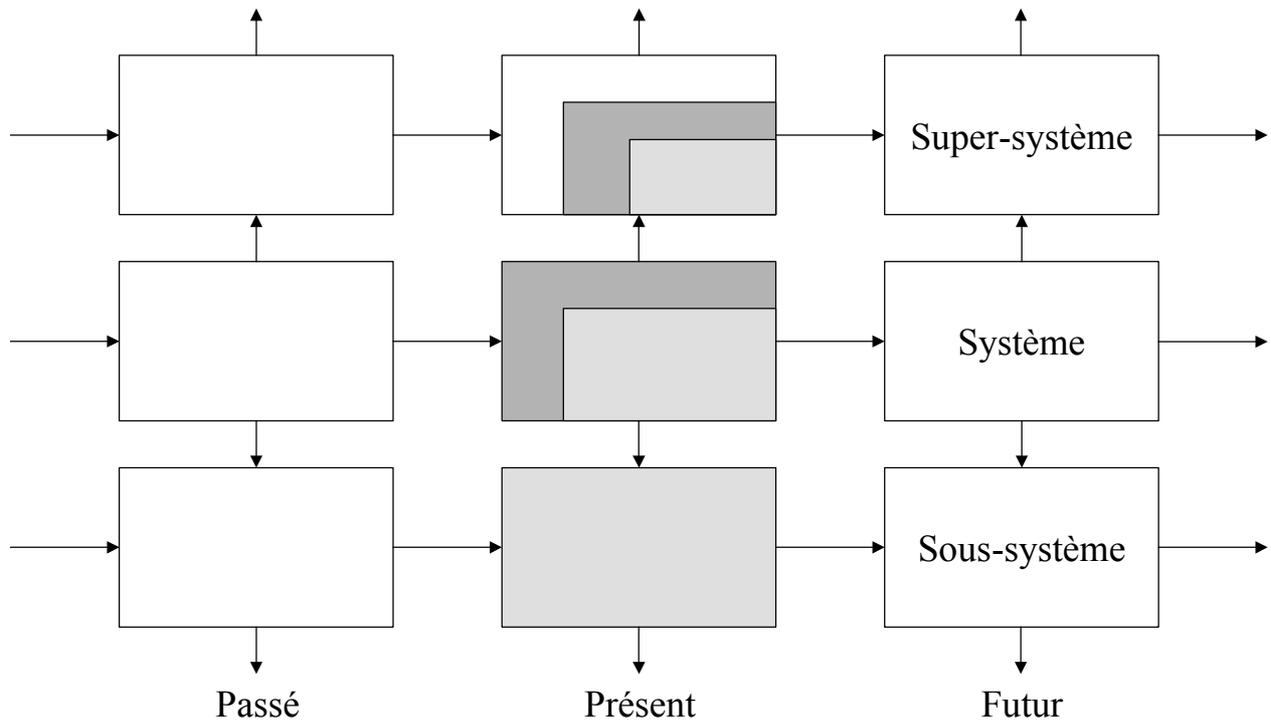


Figure II.6. Le schéma multi-écrans, selon [Altshuller, 1988]

L'utilisation des écrans permet de corriger l'habitude qu'a un ingénieur de se concentrer uniquement sur une partie du système qu'il doit concevoir, négligeant l'environnement d'évolution. De même, un ingénieur, s'il sait identifier les niveaux systémiques d'un système dans le présent, il n'utilise pas la vision systémique dans le passé, et très peu dans le futur ([Savransky, 2000]). Une utilisation systématique de l'ensemble des écrans apporte un éclairage sur les modes d'évolution du système considéré. En effet, l'utilisation des écrans relatifs au passé, permet de visualiser l'évolution suivie par le système et d'identifier les contradictions résolues par le système. On peut ainsi mieux appréhender les contradictions que la future génération du système aura à résoudre.

Enfin, Altshuller apporte une dernière dimension à l'esprit fort, qui est l'utilisation de l'anti-système ([Altshuller, 1988]). L'anti-système réalise la fonction opposée au système considéré. Ainsi l'opposé de la fonction chauffer est la fonction refroidir, de la fonction mélanger est la fonction séparer, ... ([Savransky, 2000]). Les techniques ainsi définies, permettant la réalisation de la fonction opposée, sont potentiellement intéressantes pour la synthèse du système à concevoir ; en tous cas la définition des techniques annihilant la fonction à réaliser permet de mieux comprendre les propriétés de cette fonction, et d'en définir les limites.

Dans [Khomenko, 2001], Khomenko ajoute, aux axes temporel et systémique, de nouveaux axes de description. Ces axes caractérisent la nature des informations des informations recueillies lors de la construction du modèle de problème :

- Le niveau d'objectivité, il définit le degré d'objectivité des causes des problèmes identifiés, afin d'identifier, pour s'en abstraire, les points de vue.
- L'indice de possibilité permet de caractériser, dans les objectifs à réaliser, ce la part de ce que l'on sait réaliser de ce qui est jugé comme impossible. Cette description permet de diviser les problèmes en sous-problèmes, plus faciles à résoudre.
- La fréquence d'apparition de la situation problématique considérée quantifie l'itération des situations envisagées, caractérisant l'importance du problème.
- La variation des valeurs impose l'intensification des paramètres. La caractérisation des situations extrêmes renseigne le problème sur d'éventuelles voies de résolution.
- Le niveau d'abstraction : il définit le niveau de détail de la description faite. Le niveau d'abstraction est caractérisé par le nombre de paramètres inclus dans la description.
- Les relations de cause à effet conduisent à l'identification des relations non prises en compte dans l'analyse du problème. Les relations de cause à effet permettent en outre de classer les sous-problèmes et de créer une hiérarchisation.

Le schéma de l'esprit fort est défini dans [Kucharavy, 2002], comme un méta-modèle de développement de la description de la situation problématique. Ce schéma est une réponse à la contradiction de la description des problèmes, à savoir :

- La description de la situation problématique doit être longue et détaillée, afin d'avoir suffisamment d'informations pour développer les concepts de solution.
- La description de la situation problématique doit être courte et brève, afin de ne pas consommer de ressources de temps et de personnel.

Le schéma propose donc de guider cette description de la situation problématique. En caractérisant la nature des informations à collecter, le schéma fournit un outil qui permet de capitaliser l'information nécessaire à la résolution. Il s'agit de ne considérer que cette information nécessaire pour développer les concepts de solution en un minimum de temps.

II.3.2. Les standards de résolution

Nous avons déjà formulé précédemment le fait que formulation et résolution d'un problème sont deux processus conjoints. Nous allons ici présenter un outil de modélisation des systèmes : le modèle Vépole, auquel sont associées des règles de transformation de ces modèles : les standards de résolution.

II.3.2.1. Le modèle Vépole

La modélisation Vépole est une représentation bipolaire du monde technique ([Altshuller, 1988]). Le terme Vépole vient des mots russes « Vechestvo » et « Pole » qui signifient, respectivement, substance et champ. Le modèle Vépole est donc une modélisation des systèmes techniques sous forme de substances, de champs et de leurs interactions (voir [Savransky, 2000]), dans la suite du document, nous utiliserons la terminologie « modèle substance-champ », et uniquement le terme Vépole, dans les lignes de programmation du logiciel :

- La **substance** réfère à un objet matériel, quel que soit son niveau de complexité. La substance peut ainsi être un élément simple (vis, couvercle, ...) ou un système complexe (voiture, navette, ordinateur, ...).
- Le terme **champ** est utilisé ici dans un sens très large. Y sont inclus les champs de la physique (électromagnétiques, gravitationnelle, interactions nucléaires, ...), mais également les champs chimiques, acoustiques, ... Un champ est un flux d'énergie, d'information, de force, une réaction ou une interaction réalisant un effet. La présence d'un champ sous-entend la présence d'une substance, car elle est la source du champ.

La réalisation d'une fonction passe par l'interaction entre deux substances. Cela provient du fait que pour permettre à un champ de réagir sur une substance, il doit le faire via une seconde substance ([Altshuller, 1988]).

Ainsi, pour qu'un système soit efficient, il doit être constitué au minimum de deux substances et d'un champ. La représentation triangulaire, représentée figure II.7., est la modélisation d'un système minimal, fonctionnel et contrôlable.

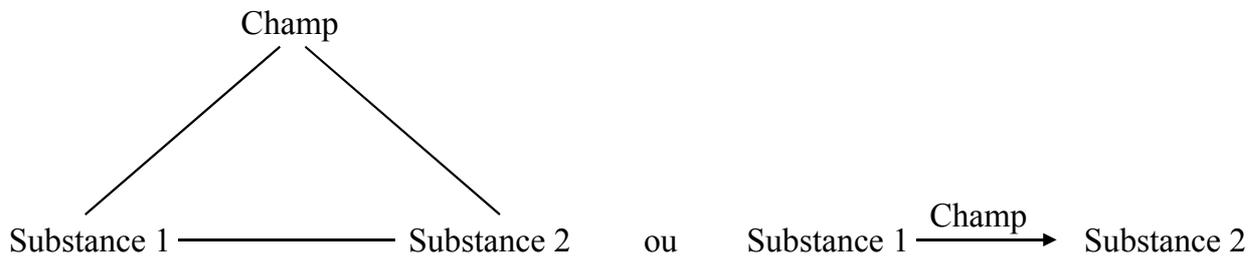


Figure II.7. Représentation minimale d'un système fonctionnel, selon [Salamatov, 1999] et [Cavallucci, 1999]

Le dernier élément de modélisation est la caractérisation des interactions qui permet de spécifier la nature des liens entre substances et champs. Les interactions sont classifiées selon qu'elles sont :

- satisfaisantes pour le système, car elles participent correctement à la réalisation d'une fonction du système ;
- néfastes pour le système, générant des désagréments ou participant à la réalisation de fonctions non souhaitées ;
- excessives, puisque ce sont des interactions utiles qui participent à la réalisation d'une fonction du système, l'intensité devant toutefois en être minimisée ;
- insuffisantes, parce que ce sont des interactions utiles qui participent à la réalisation d'une fonction du système, mais l'intensité doit en être augmentée.

A chaque typologie d'interaction correspond une représentation graphique, comme le montre la figure II.8. (cf. [Altshuller, 1980; Cavallucci, 1999; Salamatov, 1999; Savransky, 2000]).

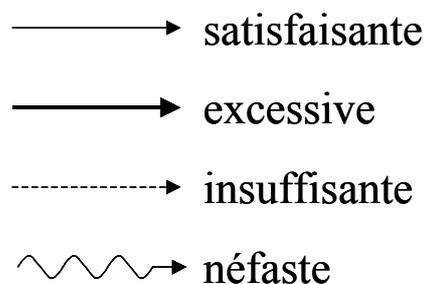


Figure II.8. Typologie des interactions

II.3.2.2. Les standards de résolution

L'intérêt de la modélisation substance-champ, exposée in supra, est de permettre la représentation, de manière simple et graphique, des problèmes. Une contradiction est issue de la difficulté de réalisation simultanée de deux fonctions ; soit qu'il n'est pas possible de réaliser simultanément deux fonctions utiles, soit que la réalisation d'une fonction utile entraîne l'apparition d'une fonction néfaste. Une contradiction fait intervenir les substances impliquées dans les fonctions mises en jeu, et les champs qui permettent la réalisation desdites fonctions. Cet ensemble de substances et de champs forme le système à considérer, en regard du problème à résoudre. Ce système est, la plupart du temps, différent du système principal considéré initialement.

Le modèle substance-champ, ainsi construit, donne une représentation graphique de la contradiction, offrant un modèle visuel du problème. L'inconvénient est que le lien entre les deux représentations, le modèle substance-champ et celui de la contradiction, n'est pas explicite. L'intérêt de la modélisation substance-champ est l'existence d'un outil pour transformer ce modèle graphique du problème en modèle graphique de solution : les standards de résolution.

Altshuller a synthétisé les méthodes standards utilisées par les inventeurs dans leurs travaux, puis a construit une dialectique de définition de ces standards. Ils sont au nombre de 76 à ce jour (cf. [Cavallucci, 1999; Savransky, 2000; Slocum, 2002; Terninko et al., 2000] et Annexe-I). L'objectif de chaque standard étant, après une modélisation du problème sous forme de modèle substance-champ, de passer à un ou plusieurs modèles de solutions types. Les standards sont construits sous la forme de recommandations, et généralement, formulés selon le schéma suivant :

Si *<Condition 1>* et *<Condition 2>*, alors *<Recommandation>*.

Les deux conditions permettent de reconnaître la typologie du problème associé au standard. Ainsi, pour un modèle de problème construit, il existe un certain nombre de recommandations permettant de construire le modèle de solution correspondant (cf. l'exemple figure II.9.).

Standard 1.2.1.:

Si < dans le vépole, des actions utile et nuisible surviennent en même Temps > et si < il n'est pas obligatoire de garder le contact direct entre elles, > alors < on résout le problème en introduisant entre ces substances une troisième substance étrangère gratuite ou assez bon marché. >

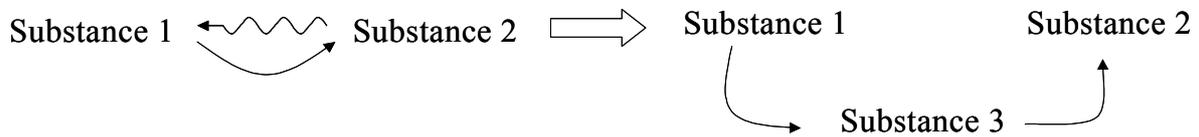


Figure II.9. Exemple de standard, issu de [Salamatov, 1999]

Les 76 standards sont regroupés en cinq classes (voir figure II.10.) qui correspondent à des catégories de typologie de problèmes. Ainsi les trois premières classes sont représentatives des différents niveaux d'évolution des systèmes. L'identification, par le modèle substance-champ, de la réalisation des fonctions impliquées, permet de caractériser le niveau d'évolution du système du problème, et de le faire alors évoluer en respect des lois d'évolution des systèmes techniques.

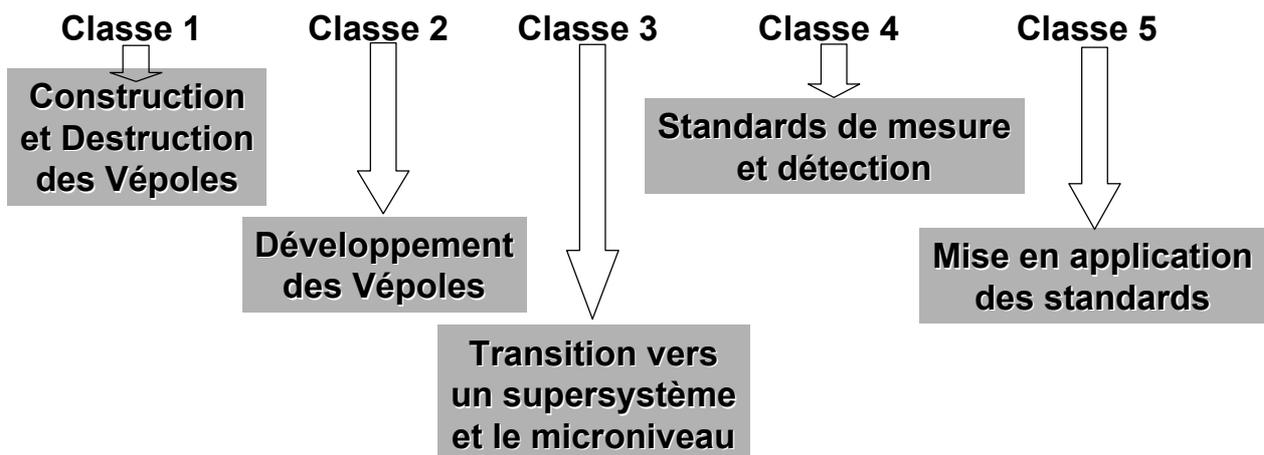


Figure II.10. Les cinq classes de standard, selon [Cavallucci, 2001]

La quatrième classe est consacrée à la réalisation de systèmes de mesure, dont la spécificité est d'avoir, pour fonction, la recherche d'un champ, qui permettra d'accéder à une information sur une substance, et d'en connaître la valeur d'un paramètre.

Enfin, la dernière classe est consacrée à la mise en œuvre des standards des quatre premières classes. Lorsque l'utilisation des standards est trop générique, cette classe permet d'affiner le mode de résolution. Par exemple, elle précise la nature des ressources à utiliser lorsqu'il est nécessaire d'ajouter une nouvelle substance (utilisation du vide) ou un nouveau champ (en privilégiant les champs les plus contrôlables).

II.3.2.3. Regards sur l'outil

Les standards sont un outil très intéressant qui présente, en soi, une méthode de résolution de problème, avec la construction d'un modèle (le modèle substance-champ) et une heuristique de modification du modèle. En outre, la synthèse des standards de résolution s'appuie sur les lois d'évolution, les classes correspondent à différents niveaux d'évolution des systèmes. La bonne construction du modèle substance-champ permet de recourir à un standard dont la mise en œuvre oriente l'évolution en accord avec les lois objectives d'évolution des systèmes techniques.

Trois critiques peuvent être formulées vis à vis de l'utilisation des standards. Tout d'abord, la corrélation entre le modèle proposé et les standards n'est pas effective, du fait de la non complétude de la représentation. Les standards font, par exemple, appel à des contraintes de modification (l'obligation de contact entre substances, entre autres) qui ne sont pas présentes dans le modèle. Par ailleurs, des incohérences de langage sont présentes et la formalisation proposée ne permet pas l'utilisation de l'outil en outil bijectif. Enfin l'existence de la classe cinq montre des limites à la classification proposée. Alors que les premières classes représentent des niveaux différents d'évolution d'un système, la classe 4 réfère à une typologie de problèmes particulière, et la classe 5 est une meta-classe qui fait référence aux précédentes. Cette classification ne peut être jugée satisfaisante.

II.3.3. ARIZ

La méthode principale de mise en œuvre des outils de la TRIZ est l'algorithme ARIZ (présenté dans [Altshuller, 1999]). Celui-ci a pour but de guider le concepteur de l'identification d'une situation problématique jusqu'à l'identification d'une solution optimale. L'algorithme inclut également une phase d'exploitation maximale de la solution, sa brevetabilité, son utilisation dans d'autres domaines, ... ; de même qu'une phase d'analyse du processus de résolution suivi (voir figure I.10.).

II.3.3.1. La genèse de la méthode

ARIZ est un programme d'actions qu'un concepteur doit suivre lorsqu'il est amené à résoudre des problèmes techniques difficiles ([Savransky, 2000]). Cet algorithme russe signifie *Algorithme pour la résolution des problèmes inventifs* ([Altshuller, 1988]). Le but de cet algorithme, tel que le définit Genrich S. Altshuller, est de fournir, aux inventeurs, des opérateurs qui guident pas à pas à la contradiction physique, et, déterminent la partie du système technique dans laquelle cette contradiction apparaît.

A l'origine de cette recherche est la constatation qu'il existe différents degrés d'inventivité et que les problèmes résolus, par une seule personne, entrent dans la catégorie de faible inventivité. Les problèmes plus complexes, s'ils sont abordés par la méthode des essais et erreurs, n'offrent que rarement des solutions satisfaisantes. Un besoin se fait ressentir : celui d'une nouvelle technologie pour la résolution de problèmes inventifs, permettant à un inventeur (seul) de résoudre des problèmes d'ordre plus complexe. Cette technologie doit être basée sur la connaissance des lois objectives de développement des systèmes techniques.

ARIZ se décompose en un certain nombre d'étapes permettant d'obtenir très rapidement une classification du problème étudié, et ainsi une vision de l'orientation à donner à la recherche de solution. Il permet, en outre, de faire évoluer un problème jugé complexe, vers une formulation qui en fait un problème simple à résoudre ([Marconi, 1998]). Il fournit une méthode formalisée de prise en compte du système considéré, ainsi que de son environnement (systèmes adjacents, ressources disponibles, super-système,...) pour aboutir à la formulation la plus pertinente possible du problème, en vue de sa résolution par l'un des outils mis à disposition par la théorie. Aujourd'hui est usuellement utilisé ARIZ 85C. Ceci sous-entend qu'il existait différentes moutures de l'algorithme. On en dénombre en effet 16 évolutions ([Altshuller, 1999]). Il faut noter que chaque modification, apportée à l'un des quelconques outils de la TRIZ, a fait l'objet de tests de validité suivant le processus : mise en condition pédagogique, études menées

par Altshuller et les experts de la théorie, utilisation en séminaires et enfin, application sur différentes situations de vie courante ([Zlotin and Zusman, 1998]). Les principales évolutions furent ainsi ([Altshuller, 1986]) :

- La première version date de 1956. ARIZ-56 n'est, en fait, qu'un ensemble d'étapes pour la résolution de problèmes. Basé essentiellement sur les comportements des meilleurs inventeurs du passé, l'algorithme ne profite pas encore de l'analyse des brevets.
- ARIZ 59 voit l'introduction d'une notion primordiale de TRIZ, le Résultat Idéal Final.
- Un tournant important de l'algorithme, en 1964, avec l'introduction de la partie : *Clarification et vérification du positionnement du problème*. Enfin, cette étape marque l'apparition de la première table des principes inventifs de résolution des contradictions techniques, sur laquelle une matrice sera constituée dès l'année suivante (ARIZ-65).
- En 1971, est introduite la prise en compte de la zone opérationnelle et de ses éléments contradictoires (posant les jalons des futures contradictions physiques). La version 71 marque aussi l'apparition des opérateurs DTC (Dimension, Time, Cost), qui proposent d'intensifier les conflits par des distorsions de taille, de temps ou de coûts pour dépasser l'inertie psychologique. Enfin la version complète de la table des contradictions est proposée, des principes d'innovations ont étant encore ajoutés (cf. Annexe-II).
- Les transformations du modèle substance-champ sont introduites en 1975.
- La version de 1977 est l'aboutissement d'un travail entamé depuis ARIZ 71. L'introduction des contradictions physiques (à un micro-niveau) dévoile la voie envisagée pour les développements futurs (cf. [Arciszewski, 1988] et Annexe-III).
- De ARIZ-82 à ARIZ-85A, nous entrons dans le développement d'un "algorithme pédagogique" qui s'appuie sur le fait que toute recommandation ou note, émise par un enseignant, se doit d'être incorporée à l'algorithme. De nouvelles étapes sont introduites telles que la définition systématique d'une « contradiction micro-physique », d'un Résultat Idéal Final 2 ou encore l'analyse du modèle de problème.
- Dans la version ARIZ-85C (voir [Altshuller, 1999] et Annexe-IV), l'analyse des ressources substance-champ a été singulièrement renforcée, de même que la

recherche d'une solution dans la direction de l'idéalité (en renforçant l'idée du vide comme la ressource la plus pertinente). Les liens entre l'algorithme et les solutions standards, les brevets d'évolution technologique deviennent plus forts. Cette version est la plus aboutie et celle traditionnellement utilisée.

Cette évolution met en évidence une algorithmisation⁵ de plus en plus poussée qui, si elle était poursuivie, aboutirait à un algorithme des plus précis des étapes de conception, algorithme qui composerait un recueil de 300 pages. Cette voie n'est pas souhaitable, car elle nécessiterait un long et fastidieux apprentissage. La question à se poser est : « Pourquoi avoir besoin de détailler autant la réalisation des concepts présents dans la méthode ? ». Nous apporterons notre propre vision de la réponse plus loin, mais voyons d'abord les éléments de réponse apportés par les experts de la TRIZ.

II.3.3.2. Les évolutions « post-Altshuleriennes »

ARIZ est considéré ([Marconi, 1998]) comme une technique avancée de la TRIZ. Il impose une définition précise de tous les éléments du problème, une utilisation itérative de tous les outils de résolution de problèmes de TRIZ, en proposant des va-et-vient, du domaine physique au domaine fonctionnel, du système au super-système et aux sous-systèmes. Outre une méthode de conception, l'algorithme s'avère être un outil pédagogique intéressant, comme le montre [Grace et al., 2001]. Altshuller lui-même préconise, dans [Altshuller, 1986], d'axer le développement d'ARIZ sur un renforcement de la fonction éducative, afin de développer plus activement les acquis de la pensée forte.

Cependant, de nombreux écrits remettent en cause cette version et ne la jugent pas suffisante pour aborder les problèmes d'innovation. La première étape d'ARIZ, « analyse du problème », qui permet de transformer une situation globale et floue en un problème technique concret à résoudre, est la plus sujette à discussions. Ivanov ([Ivanov and Bystritsky, 2000]) propose de l'élargir et de la doter d'un outillage précis, un algorithme détaillé et argumenté de nombreux commentaires et exemples. Pinyayev recommande de substituer l'étape 1.1 d'ARIZ 85 par un algorithme pour l'analyse fonctionnelle des situations inventives ([Pinyayev]). Il met en avant la difficulté du passage, en une seule étape (1.1), de la contradiction administrative à la contradiction technique, alors que trois parties de l'algorithme sont dédiées au passage de la

⁵ Les algorithmes sont ici des séquences d'étapes à réaliser avec une très forte interaction avec un utilisateur humain, l'algorithme n'est pas automatisable.

contradiction technique à la contradiction physique (12 étapes). Considérant que l'étape 1.1 est l'une des plus difficiles à réaliser, il propose un algorithme dont le propos est de situer et de faire émerger un problème inventif. Le résultat de cet algorithme est la formulation de contradictions techniques et des objectifs à atteindre pour résoudre le problème énoncé. Ces recherches vont dans le sens d'un algorithme sans cesse rallongé, le rendant non utilisable en situation de conception, soumise à des impératifs de temps.

D'autres experts critiquent la non adéquation des informations exigées dans les premières étapes du processus, par rapport à celles nécessaires à la synthèse des concepts ([Souchkov, 1998]). De plus, différentes parties d'ARIZ nécessitent des modélisations de problèmes, toutes différentes (modèles substance-champ, contradiction, Résultat Idéal Final), ne faisant apparaître ni lien évident entre toutes ces représentations, ni indications sur la manière de passer d'un modèle à un autre. Ceci induit le besoin d'une forte compétence sur la TRIZ, alliée à une forte expertise du domaine du problème à résoudre pour obtenir des solutions de qualité.

Dans [Zlotin and Zusman, 1991], les modifications à apporter à l'algorithme relevées sont :

- L'introduction de nouvelles parties pour les phases de formulation de problème.
- L'intégration de l'outil matriciel de résolution des contradictions techniques.
- La segmentation de ARIZ en micro-algorithmes, exemples et volumes de définitions.

Ces modifications poussent, à nouveau, à l'extension de l'algorithme et à le rendre ainsi de plus en plus difficile à mettre en œuvre. L'idée de segmenter en micro-algorithme permettrait de produire plusieurs voies de formulation du problème en fonction des conditions de ce dernier, mais il ne faut pas que cela se fasse au détriment de la résolution. Or, il est ici envisagé de segmenter selon les outils de résolution, ce qui ne permet pas de profiter, en toutes circonstances, de la force des outils de résolution que sont les standards ou les principes de séparation de la contradiction physique. C'est également la voie préconisée dans la construction d'ARIZ-2000 par Soderlin ([Soderlin, 2003]).

Enfin des recherches ont montré que l'algorithme est une méthode basée sur la psychologie ([Dikker et al., 1992]). Un certain nombre d'étapes ne sont pas formulées explicitement, laissant l'intuition de l'ingénieur guider la résolution. Afin d'automatiser ces étapes, il est nécessaire de les rendre explicites en identifiant les connaissances nécessaires à leur réalisation.

Pour sa part, Altshuller, a cessé de développer des méthodes et outils pour la résolution des problèmes techniques après ARIZ-85 ; il s'est centré sur la formalisation de méta-outils et méta-

méthodes pour la résolution de problèmes universels, le précisant dans [Altshuller, 1986]. De tels programmes se sont développés, notamment le projet Jonathan Livingstone à Minsk ([Khomenko, 2001; Khomenko and Sokol, 2000]).

II.3.3.3. Le besoin d'un modèle unique

Notre position par rapport à une évolution possible de ARIZ est toute différente. Comme mentionné plus haut, avant de relever d'éventuels manques sur cet algorithme, nous préférons y voir les clés du succès et comprendre pourquoi, alors que tout semble y être présent, l'algorithme ne remplit pas sa tâche comme il le devrait. ARIZ est en effet une méthode complète, en ce qu'elle comporte :

- L'identification des éléments informationnels à capturer sur la situation problématique.
- La construction, sur la base de ces informations, de modèles de problèmes génériques permettant une meilleure compréhension des difficultés de résolution rencontrées.
- L'existence d'outils de construction progressive de la solution.
- Les modes d'interprétation des concepts de solution et les voies de réalisation.

Dès lors, comment se fait-il que cet algorithme ne soit pas opérationnel ? ARIZ est décrit comme « *une méthode basée sur la psychologie* » pour laquelle nombre d'étapes ne sont pas « *formulées explicitement mais laissées à l'intuition de l'ingénieur* » ([Dikker et al., 1992]). Nous pensons que la difficulté majeure est liée à l'existence de plusieurs modèles de représentation des problèmes (le modèle substance-champ et la contradiction physique) et à une co-habitation délicate entre eux. Alors que ces modèles sont des points de vue différents d'un même objet, ayant un objectif commun, comment se fait-il qu'il semble y avoir divergence dans la construction ? En fait, il n'y a pas de divergence, mais le manque de rigueur dans la formalisation des modèles, lié aux ambiguïtés sémantiques relevées dans les standards, crée un flou perturbateur.

Nous pensons donc que, pour accroître l'efficacité de ARIZ, il est nécessaire de partir des concepts présents et de formaliser les concepts de la méthode de description des problèmes afin

de supprimer les ambiguïtés. Une telle ontologie⁶ permet de définir un modèle unique de représentation du problème, qui soit valable, à la fois pour l'outil de résolution des contradictions physiques, et pour l'utilisation des standards. Comment construire une telle ontologie ? Nous le détaillons ultérieurement, mais notre vision est de nous appuyer sur l'existant, à savoir l'algorithme ARIZ qui identifie les éléments à capitaliser pour la construction de la contradiction physique, et le texte des standards, très riche, qui décrit l'ensemble des situations de problèmes génériques pouvant être rencontrées.

⁶ Nous utilisons ici la définition de l'ontologie telle que la donne [Kitamura and Mizoguchi, 2002] Kitamura, Y. and Mizoguchi, R. (2002) Ontology-based modeling of product functionality and use. Part 1: functional-knowledge modeling. *The Third International Seminar and Workshop Engineering Design in Integrated Product Development (EDIProD 2002)*, Lagow, Poland, October 10-12. : l'ontologie fournit les concepts fondamentaux pour capturer le mode ciblé, et fournit un vocabulaire pour la description des connaissances.

II.4. Des outils informatiques associés

Nous tenons à présenter ici, succinctement, les divers outils informatiques développés sur la base de la TRIZ. Ce, afin de montrer les voies suivies et les choix de développement réalisés jusqu'ici.

II.4.1. TechOptimizer ([Invention Machine, 1999])

TechOptimizerTM est un logiciel, basé sur la TRIZ, proposé par la société Invention Machine. Le logiciel propose d'aborder un projet de conception par 7 modules différents, tous basés sur la même approche ([Crubleau, 2002]). Le module principal est l'analyse produit qui permet l'analyse fonctionnelle des produits. Le module est basé sur l'identification des fonctions principales des composants du système et la caractérisation de leurs interactions. Cette caractérisation permet d'identifier les liaisons présentant des caractères néfastes, l'identification du problème se faisant ainsi par positionnement morphologique des défauts (voir figure II.11.).

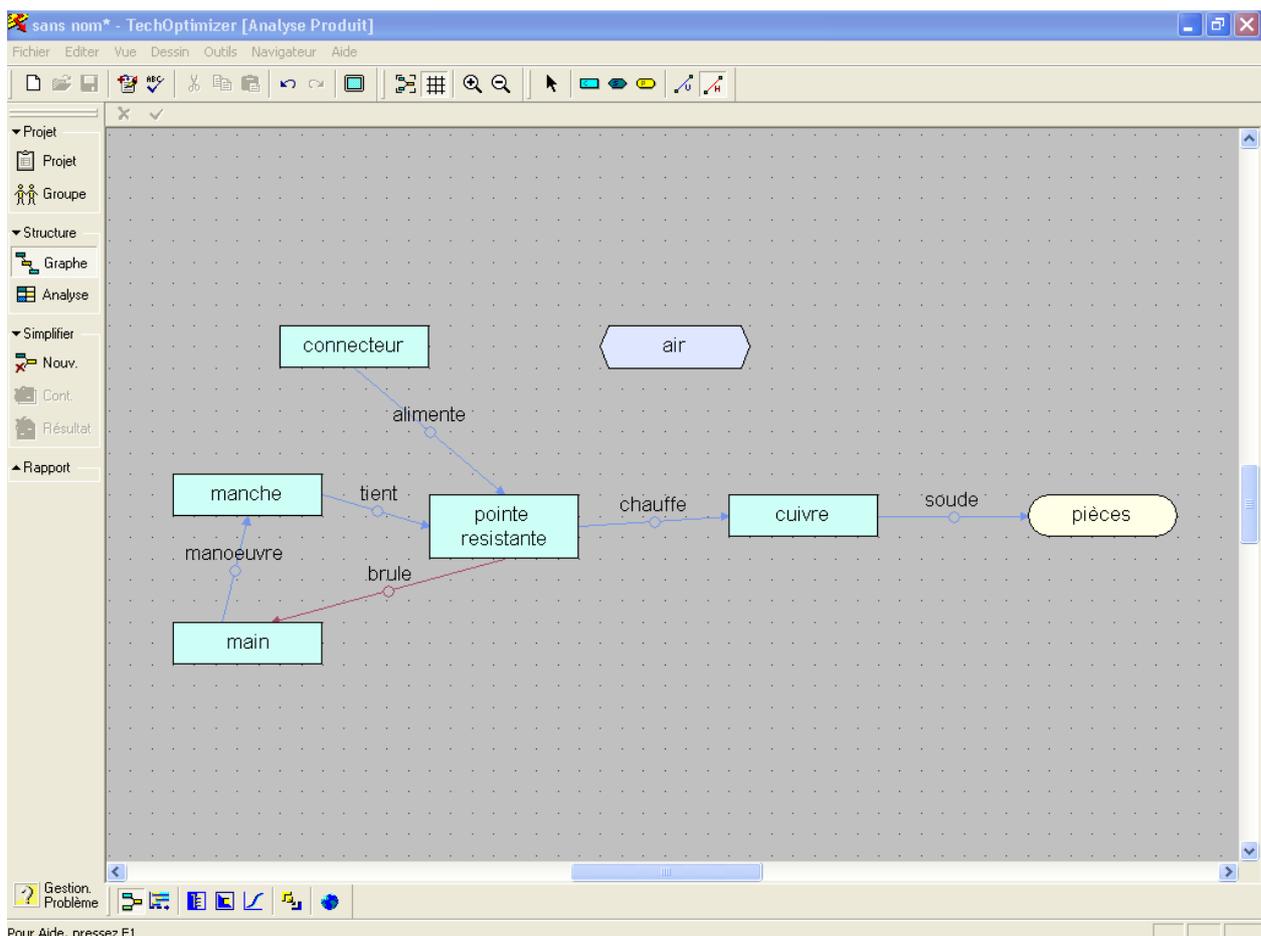


Figure II.11. Le logiciel TechOptimizer

L'approche ainsi proposée, par le logiciel, est proche d'une analyse fonctionnelle, et le rapprochement avec l'analyse systémique proposée par la TRIZ n'est pas présent. L'identification des problèmes se fait par une analyse exhaustive des liens entre les éléments, le processus convergent n'étant, dès lors, pas présent.

En revanche, les outils de résolution des problèmes sont en partie inclus dans le logiciel. On retrouve ainsi :

- La matrice de résolution des contradictions techniques, dont la formulation a quelque peu évolué, suite à une mise à jour résultant d'une étude poursuivie et systématisée des brevets.
- Les lois d'évolution se retrouvent à travers la formalisation de tendances d'évolution des systèmes techniques, agrémentées de nombreux exemples, permettant une bonne compréhension des modes d'évolution.
- L'utilisation des effets est l'outil le plus présent et sans doute l'intérêt majeur du logiciel. Le module « effets » constitue une base de données de l'ensemble des effets apparus, tous illustrés d'exemples applicatifs permettant une bonne compréhension et une mise en œuvre aisée.

Le logiciel TechOptimizer permet d'accompagner une équipe de recherche et développement durant tout le processus de la conception conceptuelle ; il permet en outre une génération automatique de rapport très appréciable, qui formalise la démarche suivie. Malheureusement cette démarche ne s'appuie que partiellement sur celle de processus convergent telle que la TRIZ la préconise. Mais la non utilisation des standards de résolution et des principes de séparation de la contradiction physique montre que le processus de capitalisation des informations, de modélisation du problème, n'est pas celui de la TRIZ.

II.4.2. Innovation Workbench ([Ideation, 2000b])

Innovation WorkbenchTM est un logiciel conçu par la société Ideation. Le processus de résolution de problème qu'il propose est basé sur le méthodologie Ideation/TRIZ. Cette méthode s'appuie fortement sur ARIZ, elle peut être vue comme une informatisation de l'algorithme et repose sur cinq phases principales qui sont :

1. Innovation Situation Questionnaire (ISQ)
2. Problem Formulation
3. Prioritize Directions
4. Develop Concepts
5. Evaluate Results

La démarche s'appuie, en outre, sur des outils analytiques développés, selon [Ideation, 1999], pour supporter les étapes manquantes du processus de la TRIZ. Ainsi, « *Innovation Situation Questionnaire* » est proposé pour faciliter la compréhension et la documentation du problème étudié. « *Problem Formulation* » est utilisé pour analyser un problème et développer de façon exhaustive l'ensemble des directions potentielles d'innovation. Ce dernier point n'abonde pas dans le sens souhaité par le processus convergent de la TRIZ.

Comparons maintenant l'outil ISQ à la démarche proposée par ARIZ, car il est proposé, dans [Terninko et al., 1998], d'utiliser ISQ en lieu et place des premières phases d'ARIZ. Il existe de nombreuses similitudes entre les deux méthodes (voir la figure II.12. pour la méthode ISQ).

De nombreuses étapes sont ainsi similaires entre les deux approches. On retrouve notamment :

- la définition du système,
- de son environnement,
- des ressources,
- l'identification et la caractérisation des fonctions,
- le recours à l'idéalité.

Des étapes ont été ajoutées dans la synthèse de ISQ ; apparaissent ainsi :

- la définition de l'historique de l'apparition du problème,

- l'identification de systèmes rencontrant des problèmes similaires,
- la définition non pas d'un problème, mais d'un ensemble de problèmes à résoudre.

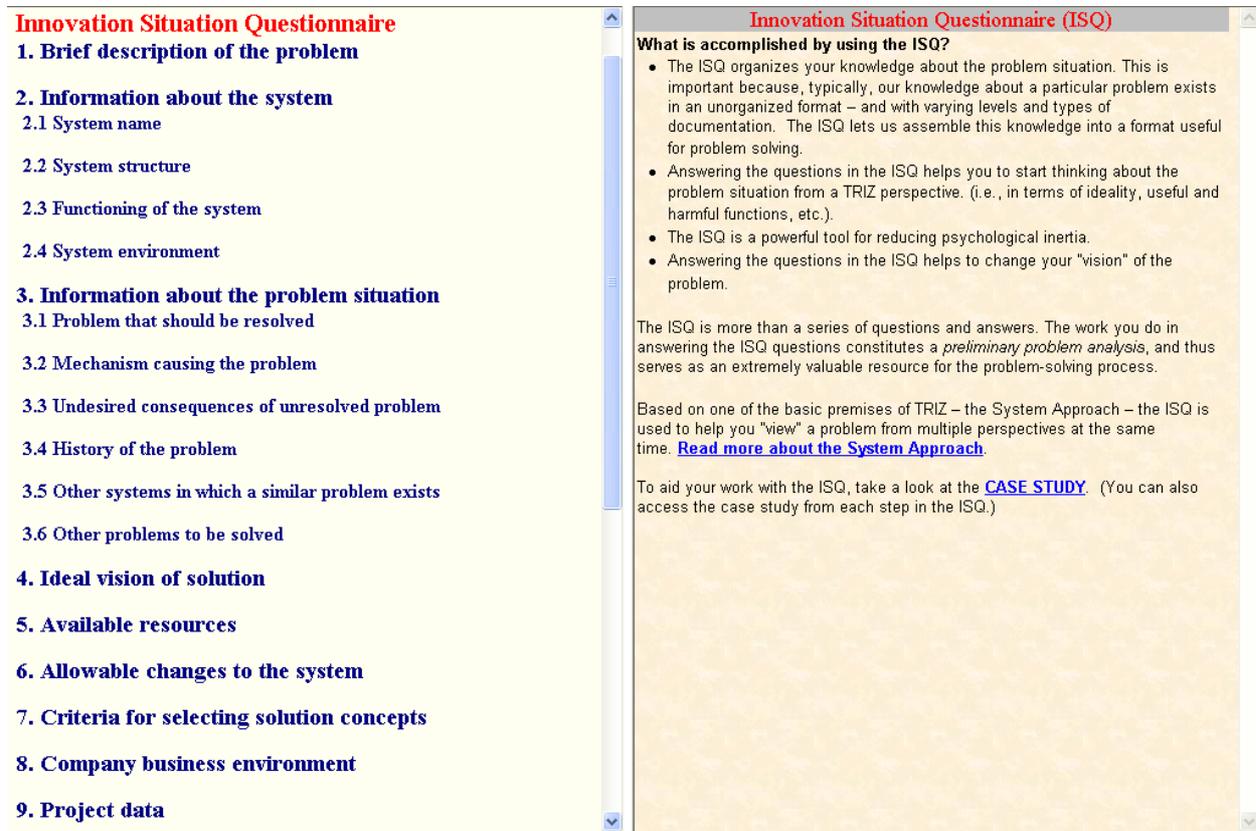


Figure II.12. La méthode ISQ, selon [Ideation, 2000b]

A première vue, ISQ semble plus complet que les premières phases d'ARIZ, et c'est le but avoué par Ideation. Mais la mise en œuvre de la méthode semble moins évidente, car les étapes sont moins formalisées que dans ARIZ ; en effet, il est nécessaire de s'appuyer sur l'aide et les exemples fournis pour réaliser correctement chaque étape.

IWB est un outil plus proche de la TRIZ, mais son objectif est de s'appuyer sur l'outil informatique pour faciliter la capitalisation progressive des connaissances. La formalisation des concepts n'est pas plus poussée que dans la version « papier », mais le logiciel s'appuie sur une base d'exemples pour faciliter la mise en application. La génération des concepts de solution se fait par une liste de choix incrémentaux, basés sur diverses voies de résolution ; l'utilisation de l'analogie n'est pas outillée, et, par ailleurs, le modèle construit est un modèle de relations de cause à effet fonctionnel, n'intégrant pas la notion de système.

II.4.3. Improver ([Ideation, 2000a])

ImproverTM est un logiciel conçu par la société Ideation. Ce logiciel contient une version abrégée du système d'opérateurs, inclus dans le logiciel IWB, pour la génération des concepts de solution. Il est conçu dans le but d'améliorer les paramètres techniques d'un produit ou d'un process ([Zlotin et al., 2000]).

Improver est un outil qui implémente une procédure systématique pour le développement de produits ou process innovants. La démarche est constituée de trois étapes :

1. La documentation du problème : elle s'appuie sur une description brève et libre de la situation. Cette documentation n'est pas outillée.
2. Le développement de concepts de solution
3. L'évaluation des résultats

Le développement des concepts de solution s'appuie sur le système des opérateurs (cf. figure II.13.). Le système des opérateurs est un réseau de solutions caractérisées par les objectifs de l'étude menée. Le choix de voies de solutions se fait au fur et à mesure, en fonction des objectifs visés.

Cet outil offre (d'après [Zlotin et al., 2000]) :

- une exhaustivité de recommandations pour un problème donné,
- un nombre important d'idées générées,
- des recommandations spécifiques et précises permettant de développer de nombreux concepts,
- un accroissement de l'aire de recherche de solutions.

La vision d'un processus convergent est une fois encore oubliée, et l'outil informatique est utilisé à seule fin de concentrer un nombre important de concepts potentiels et de pouvoir naviguer aisément parmi ces concepts. En outre, la recherche de solution se fait sans modélisation préalable du problème. Cela suppose l'existence d'un objectif de conception correctement formulé, et l'identification claire des difficultés inhérentes à la situation problématique.

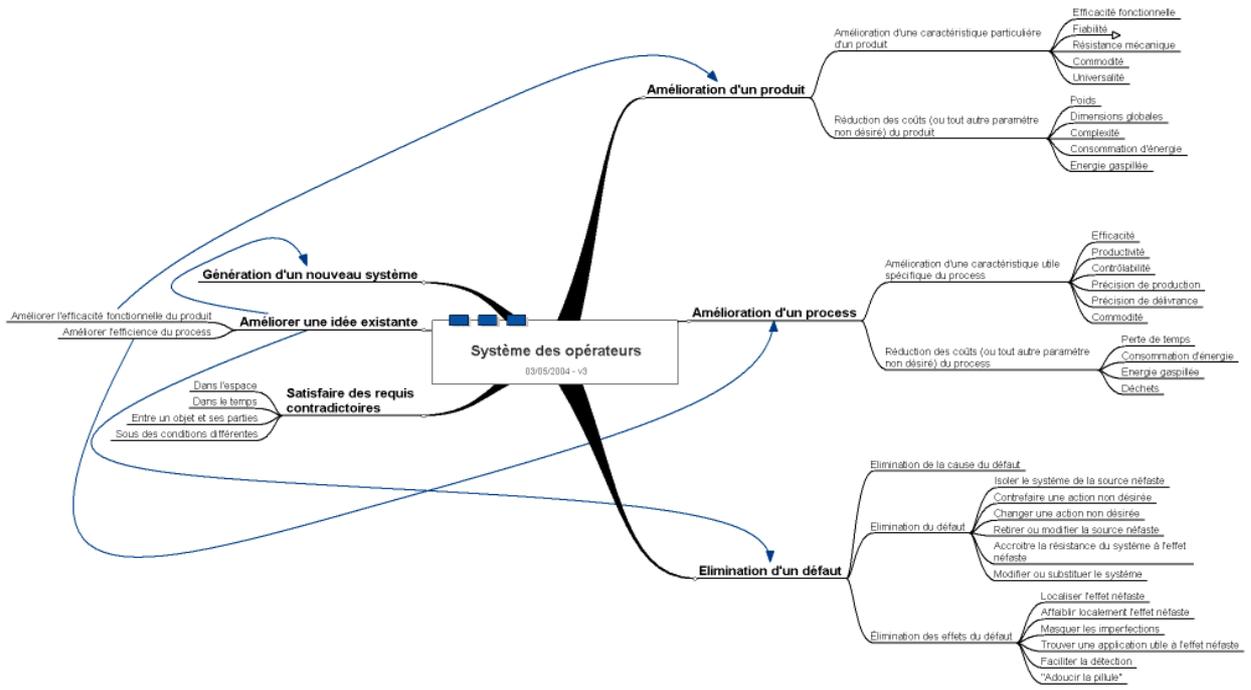


Figure II.13. Le système des opérateurs, selon [Ideation, 2000a]

II.4.4. Conclusion sur les outils informatiques

Nous allons ici présenter les apports et limites des logiciels présentés ci-dessus. Nous compléterons d'abord cette description par des logiciels dont la portée est moindre mais que nous souhaitons malgré tout citer.

On peut également citer, parmi les offres, le logiciel TriSolver ([TriSolver, 2002]) qui a informatisé ARIZ, en restant fidèle à sa formulation, et propose l'intégration des outils de résolution. Toutefois, ce logiciel n'apporte pas d'amélioration aux versions « papier », ne serait-ce l'automatisation de la génération de compte-rendus, et la facilité de modifications des étapes précédentes. Des modifications de formulation sont apportées sur les lois d'évolution et les standards, mais il n'y a pas de travail de recherche de construction d'un modèle générique permettant de passer de l'un à l'autre.

Autre produit d'Ideation, Knowledge Wizard ([Ideation, 2000c]) supporte la génération de concepts en intégrant les données de marketing, la validité des objectifs, la relativisation de la situation. L'approche est pour le reste identique à celle de IWB.

II.5. Conclusion

A présent, afin de formaliser les apports et limites de chaque outil informatique, au regard de la théorie, nous reprenons la démarche déjà utilisée pour la comparaison des méthodes de conception de systèmes techniques. Nous proposons deux rosaces reprenant les principaux points de la théorie et y caractérisons la prise en compte puis le degré de formalisation de chaque outil.

Sur la figure II.14., nous constatons l'évidente complémentarité des outils informatiques tout comme la non existence d'un outil universel. Ces remarques rejoignent celles déjà émises pour la comparaison des méthodes de conception. On peut remarquer que les outils informatiques apportent une réponse partielle à la couverture de la TRIZ, notamment au niveau de la prise en compte des outils de résolution. Ceci est du au fait que les sociétés ont privilégié le développement d'outils qui leur sont propres, sur la base des outils de la TRIZ, évitant de reprendre ceux-ci à l'identique. Elles éludent ainsi le problème d'un modèle commun à l'ensemble des outils. L'exemple des opérateurs dans Improve le montre bien.

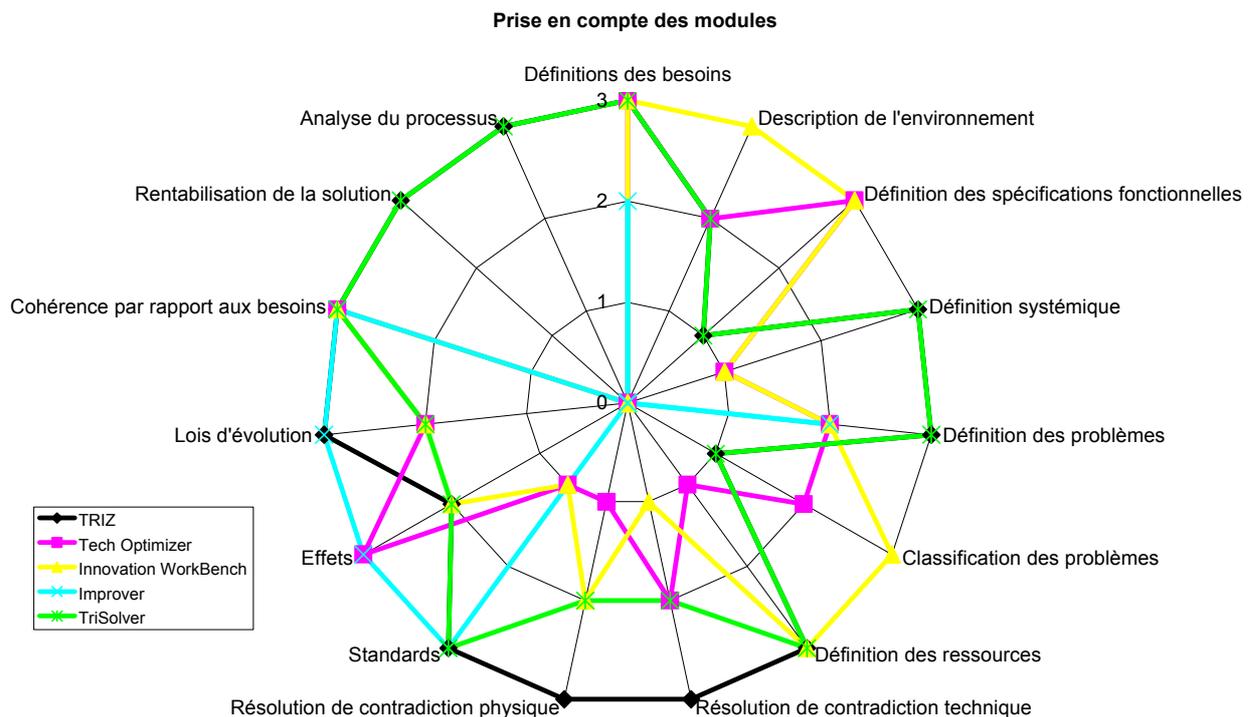


Figure II.14. Prise en compte des modules par les outils informatiques

Il pourrait être intéressant de construire les liens entre ces outils, afin de permettre une utilisation conjointe aux diverses étapes du processus. Malheureusement, les modèles utilisés ne sont pas les mêmes et la capitalisation des informations n'est pas toujours construite avec le

même objectif. Par exemple, les fonctions ne représentent pas la même chose dans IWB et dans TriSolver. De plus, une telle recherche n'est pas envisageable, car elle nécessiterait l'acquisition de l'ensemble de l'offre logicielle.

La figure II.15. montre le degré de formalisation de chaque module par les outils informatiques. On peut constater que, si les modules sont globalement pris en compte, le degré de formalisation, et ainsi la facilité de mise en oeuvre de ces modules, peuvent être accrus.

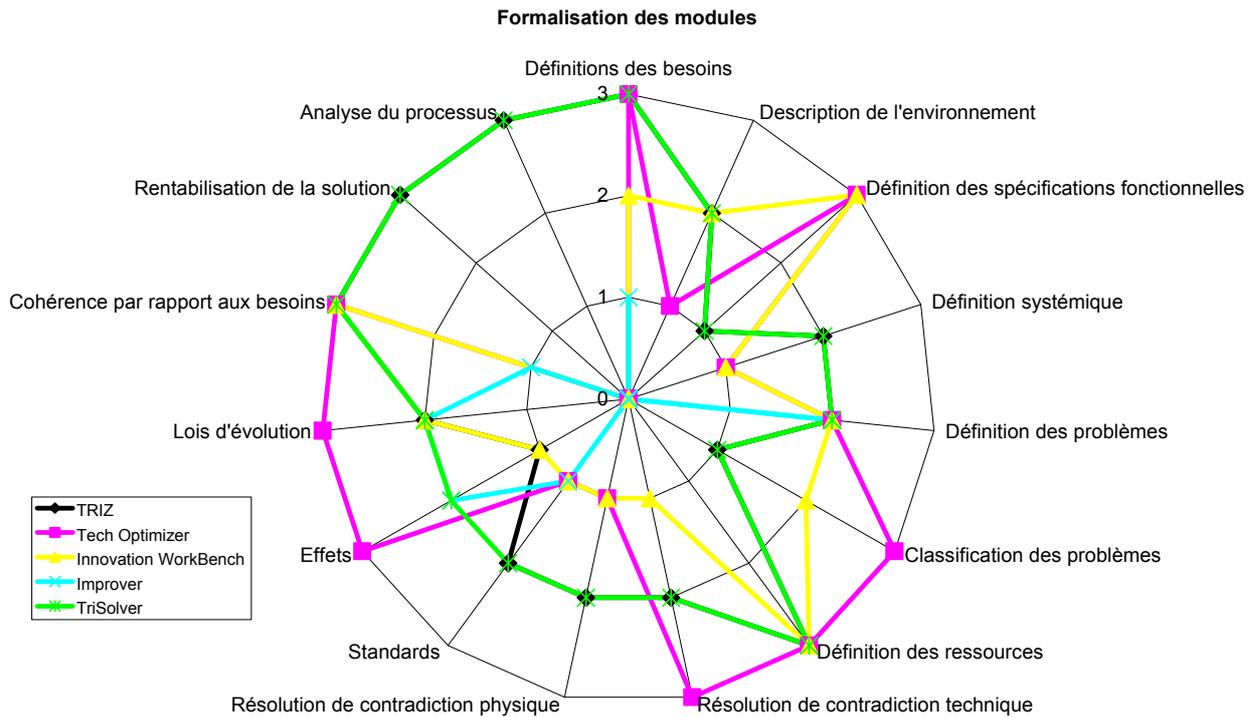


Figure II.15. Formalisation des modules par les outils informatiques

Deux manques notables sont détectés :

- la description systémique du problème et la définition du cœur du problème,
- l'utilisation des principes de séparation des contradictions physiques et les standards de résolution.

Ces deux manques vont en fait de pair, puisque l'on a, d'une part, la construction du modèle de problème, par la vision du système impliqué dans le problème et par la recherche du paramètre source de conflit ; et, d'autre part, la transformation de ce modèle de problème en un modèle de solution. Re-formulation et résolution du problème, deux étapes parallèles qui sont indissociables. Nous proposons donc de travailler à accroître la formalisation de la construction

d'un modèle générique de problème, permettant une mise en oeuvre aisée des outils de résolution que sont les principes de séparation de la contradiction physique et les standards de résolution.

Pour cela nous avons recours aux outils et méthodes de formalisation des connaissances de l'Intelligence Artificielle. Le chapitre suivant présente diverses approches pour la construction de méthodes de résolution de problème et la formalisation de connaissances expertes.

Au regard des deux chapitres, nous pouvons formuler clairement les apports issus de la TRIZ qui font de cette théorie une approche prometteuse pour la conception inventive :

- la TRIZ centre la conception sur l'élicitation et la résolution de problèmes ;
- la TRIZ propose des techniques d'identification du cœur du problème, afin de guider le passage de l'expression floue d'un besoin à la formalisation du problème technique à résoudre ;
- la TRIZ s'appuie sur le raisonnement par analogies pour proposer des modes génériques de transformation des modèles de problèmes en nouveaux modèles permettant de se rapprocher de la solution spécifiée par un processus convergent.

Malheureusement, cette théorie souffre de certains manques, qui nuisent à la fois à sa reconnaissance scientifique et à sa bonne implémentation. D'un point de vue scientifique, les reproches portent sur la mauvaise compréhension des liens entre les éléments, techniques et outils. Cette mauvaise compréhension est due à un manque de démarche scientifique dans la synthèse de ces éléments. Afin d'y pallier, nous proposons de :

Formaliser les éléments constitutifs de la TRIZ pour définir les concepts inhérents à la théorie et expliciter les liens entre ces concepts.

D'un point de vue industriel, les apports de la TRIZ sont reconnus, de même que les difficultés liées à sa mise en application, et donc à son intégration en milieu industriel. Il est nécessaire d'accroître l'applicabilité des techniques et outils de la TRIZ afin d'en faciliter l'intégration, pour cela nous proposons de :

Construire un outil de conception assistée par ordinateur, pour guider l'utilisateur dans la mise en œuvre des techniques de la TRIZ.

Du fait d'une formalisation relativement avancée des outils de transformation des modèles de problèmes, du fait de la difficulté de construire proprement un modèle de problème selon les

cadres de la TRIZ, et, enfin, du fait de la non faible intégration des techniques de formulation de problèmes dans les outils informatiques basés sur la TRIZ, nous proposons de :

Centrer nos travaux sur la construction d'un modèle de problème, selon les cadres de la TRIZ, permettant de passer de l'expression d'un besoin flou à l'identification de la cause technique du problème et permettant le recours aux outils de transformation des modèles de problèmes de la TRIZ.

III. CONSTITUTION D'UN MODELE DE FORMULATION DES PROBLEMES PAR LES OUTILS DE L'INTELLIGENCE ARTIFICIELLE

Dans ce chapitre, nous présentons le cœur de notre contribution, en illustrant, dans un premier temps, la manière dont l'Intelligence Artificielle approche la problématique de la résolution des problèmes. Cela donne lieu à la description des typologies de problèmes et de connaissances. Nous explicitons, sur cette base la démarche de modélisation que nous avons suivie.

Dans un second temps, nous présentons la modélisation entreprise, en détaillant chaque étape. Ces étapes ont été réalisées grâce au recours à des outils, logiciels, issus de des travaux en Intelligence Artificielle sur l'extraction et la modélisation des connaissances.

III.1. Apports de l'Intelligence Artificielle pour la résolution de problèmes de conception

III.1.1. Les approches de la résolution de problème

III.1.1.1. La problématique de la résolution de problèmes, une problématique sur les connaissances

« Tout processus de résolution de problèmes est fondé sur un ensemble de compétences très variées : compréhension du problème, [...] analyse des contraintes ainsi que des ressources disponibles, élaboration de stratégies et planification d'actions, exécution et suivi des actions planifiées et enfin jugement sur la qualité du processus pouvant conduire à d'éventuelles remises en cause. Cet ensemble de 'qualités' fait de la résolution de problèmes une des activités humaines les plus intellectuelles, et à ce titre, un sujet d'exploration privilégié pour l'intelligence artificielle. » [Caplat, 2002]

Guy Caplat pose ainsi tout l'intérêt, et également toute la complexité, de l'étude du processus de résolution de problèmes pour les disciplines de l'intelligence artificielle. Cette définition insiste, en outre, sur l'objet d'étude : ce n'est pas tant le résultat du processus de résolution, mais bien la procédure, la méthode de résolution, qui doit être explicitée. Caplat précise dans cette définition les phases et compétences impliquées dans ce processus ; il donne

également un point de vue qui en permet une automatisation informatique, but de la formalisation des actions planifiées. L'objet d'étude de l'Intelligence Artificielle ainsi posée, revenons brièvement sur sa genèse.

C'est en 1956 que naît l'intelligence artificielle, avec la proposition faite par Herbert A. Simon et Alan Newell, d'un programme, *Logic Theorist*, destiné à démontrer des théorèmes mathématiques ([Dortier, 2001]). Lorsqu'un mathématicien est amené à résoudre un problème, il dispose de données, d'axiomes et de théorèmes, mais pas d'une procédure lui indiquant les étapes à franchir jusqu'à l'obtention du résultat. Or, la construction d'un outil de résolution automatique de problèmes se base sur l'existence d'une telle procédure. Il est nécessaire de construire cette procédure, et pour Caplat cela revient à identifier, représenter des objets symboliques de raisonnement, et à disposer d'un mécanisme enchaînant l'exploitation des règles de calcul qui les mobilisent.

La compréhension et la formalisation des modes de raisonnement sont depuis longtemps des sujets de recherche en Intelligence Artificielle, qui a pour objet l'étude du raisonnement symbolique. De nombreuses études, notamment psychologiques, portent sur le sujet, avec des approches très variées, allant de l'explication peu formalisée, comme la psychologie de la pensée ([Dewey, 1910] qui montre les liens existant entre mode de représentation et objectifs de la pensée, insistant sur l'importance du point de vue), ou la psychologie cognitive ([Chosson, 1975] qui définit ce qu'est la pensée créative et s'appuie sur cette définition pour proposer des modes d'entraînement de la pensée), ou très formalisées, comme les approches de construction de systèmes experts.

Nada Matta ([Matta, 1995]) rappelle les différentes visions d'une méthode de résolution de problèmes, dans le champ de l'Intelligence Artificielle, et plus spécifiquement de l'acquisition des connaissances. Pour Clancey ([Clancey, 1993]), la résolution de problèmes est un processus inférentiel conduisant à la construction d'un modèle représentant une situation spécifique. Ce processus impose de définir une architecture relationnelle et fonctionnelle des connaissances du domaine. Cette vision de la résolution de problème comme spécifique au domaine du problème est contestée dans [McDermott, 1990]. Pour McDermott, une méthode de résolution de problèmes est une description, indépendante du domaine, de la démarche à suivre pour atteindre une solution. La méthode ne dépend pas du domaine du problème mais de sa typologie. De même, dans [Fensel and Motta, 1998], les méthodes de résolution de problème sont reconnues comme des composants de raisonnement, indépendants du domaine. Les composants sont des patrons de comportement pouvant être utilisés quelle que soit l'application.

[Matta, 1995] considère « *qu'une méthode de résolution de problèmes fournit des mécanismes, pour donner une solution à un problème.* » Cette description implique que :

- la méthode est spécifique à une application donnée ;
- la solution peut être exprimée en un modèle d'une situation particulière ;
- le contrôle permet de définir les mécanismes d'inférence et leur agencement, donnant, en outre, les directives de résolution.

Ces différentes visions permettent de définir les tenants des méthodes de résolution de problèmes. En revanche, la formalisation de ces méthodes, la manière d'identifier les mécanismes permettant d'atteindre les buts décrits, n'est pas précisée. L'état de l'art sur la capitalisation des connaissances, et notamment dans le cadre de la résolution de problèmes, est un ensemble hétérogène de modes de réponse variés, et spécifiques à chaque application. Longtemps, les recherches se sont focalisées sur l'identification et la spécification de méthodes de résolution de problèmes dans divers domaines, aboutissant à la spécification de nombreux langages associés ([Fensel and Motta, 1998]). [Matta, 1995] et [Studer et al., 1998] listent les différentes approches pour rendre explicite une méthode de résolution de problèmes, entre autres :

- La méthode des tâches génériques [Chandrasekaran and Johnson, 1993] propose un processus de raisonnement orienté sur les tâches. Chaque processus de résolution de problème met en œuvre un certain nombre de tâches génériques (classification hiérarchique, abstraction d'état, acquisition d'informations dirigée par les connaissances, synthèse d'objets par sélection et affinement de plans, vérification d'hypothèses et abduction d'hypothèses explicatives).
- Les approches à limitations de rôles [Marcus, 1988] réduisent l'acquisition et la modélisation des connaissances, qui constituent les systèmes de résolution de problème à base de connaissances, à l'identification d'un outil approprié à une tâche spécifique.
- L'approche Common-KADS [Valente et al., 1998] est un cadre de modélisation qui repose sur l'existence de composants génériques. La base des composants génériques est plus vaste et formée d'inférences donnant des indications sur les moyens à utiliser pour mener à bien les tâches.
- Protege [Musen et al., 2000] est une méthode à limitations de rôles qui se situe en amont. Il permet la construction d'un modèle de l'application et de sa méthode de

résolution de problèmes. La définition de stratégies d'acquisition peut ensuite être implémentée.

Fensel et Motta notent que cette variété dévoile le manque d'une base théorique claire, relevant, entre autres, les difficultés suivantes :

- difficulté de caractériser le champ d'application couvert par une collection de méthodes de résolution de problèmes ;
- la comparaison et la différenciation des méthodes de résolution de problèmes, affectées à des tâches différentes, ne sont pas explicites ;
- manque de lisibilité de la démarche de mise en œuvre d'une méthode de résolution de problèmes.

Pour répondre à ces limites, ils proposent un cadre de caractérisation des méthodes de résolution de problèmes par :

- l'engagement du problème (*problem commitments*) : spécification du positionnement des connaissances par la typologie des problèmes pouvant être résolus par la méthode ;
- le paradigme de résolution de problèmes (*problem-solving paradigm*) : description de haut niveau qui spécifie la structure des connaissances et fournit le mode de décomposition des tâches. Par exemple, le paradigme *génère et teste* ;
- les hypothèses du domaine (*domain assumption*) : elles imposent des contraintes à la formalisation des connaissances du domaine, afin de permettre l'instanciation d'une méthode de résolution de problèmes d'une application particulière.

La caractérisation proposée permet de définir clairement les limites et le champ d'application d'une méthode de résolution de problèmes. Elle met en évidence la séparation entre procédure de résolution et connaissances du domaine, tout en pointant la nécessaire adaptation entre les deux. Cette distinction pose la question du développement de processus de raisonnement génériques, non dépendants d'une application. Pour Schreiber, la formalisation de méthodes de résolution de problèmes passe par une description suffisamment générique de processus de raisonnement, de manière à les rendre indépendants de tout domaine d'application. Il base la caractérisation des raisonnements sur la typologie des problèmes résolus (l'engagement du problème de Fensel et Motta), et propose, dans [Schreiber, 2000], de spécifier le paradigme de résolution de problèmes selon les structures d'inférence et de contrôle. La structure

d'inférence caractérise les blocs de construction du processus de raisonnement, tandis que la structure du contrôle spécifie la stratégie de ce raisonnement.

On reconnaît dans cette classification des connaissances une manière de traiter indépendamment les « Quoi ? » des « Comment ? ». Anderson propose, dans [Anderson et al., 2001], de décomposer les connaissances en quatre typologies :

- les connaissances factuelles sont les éléments de connaissance de base à posséder pour être familier d'un domaine et y résoudre des problèmes ;
- les connaissances conceptuelles structurent les connaissances factuelles en créant et explicitant les interrelations. Les catégorisations, principes ou modèles sont des connaissances conceptuelles ;
- les connaissances procédurales correspondent aux savoir-faire et aux méthodes, mais intègrent également la connaissance des critères d'utilisation des aptitudes, méthodes, algorithmes et techniques ;
- les connaissances méta-cognitives sont des connaissances génériques sur la cognition. Elles représentent la connaissance et la conscience des stratégies d'utilisation des savoir-faire.

Les connaissances factuelles et conceptuelles sont respectivement les briques de connaissance élémentaires du domaine et les liens organisant ces briques. Cette catégorisation peut être ramenée à considérer que les connaissances factuelles et conceptuelles concernent les « QUOI ? » tandis que les connaissances procédurales et méta-cognitives se rapportent aux « COMMENT ? » ([Mayer, 2002]).

Rechenmann, dans [Rechenmann, 1995], a donné une classification similaire des connaissances, également en quatre niveaux :

- les connaissances descriptives qui portent sur les entités du domaine et sur les liens qui les organisent ;
- les connaissances comportementales faisant référence au comportement de l'entité réelle ;
- les connaissances méthodologiques qui sont des connaissances procédurales afin de créer, compléter et caractériser les entités dans le contexte de la résolution d'un problème complexe ;

- les connaissances contextuelles, indispensables, qui vont aider à appréhender la base de connaissances – sur les sources de connaissances (typiquement des articles de revues ou des monographies), les objectifs de la modélisation et les choix de représentation.

Les descriptions de Rechenmann et de Mayer sont assimilables sur les trois premiers niveaux de connaissances : factuelles et descriptives, conceptuelles et comportementales, procédurales et méthodologiques. En revanche, Rechenmann ne propose pas d'intégrer un niveau supérieur concernant la conscience des connaissances et stratégies, mais plutôt de contextualiser les connaissances. Toutefois, il semble que l'objectif des niveaux méta-cognitif et contextuel soit identique, à savoir, faciliter l'apprentissage et une bonne capitalisation ; à la nuance près, cependant, que le niveau méta-cognitif est orienté sur l'utilisateur des connaissances, alors que la contextualisation favorise le partage des connaissances.

Nous voyons que la connaissance est au cœur de la résolution de problèmes et qu'est d'importance la distinction entre, d'une part les connaissances liées à un domaine et, d'autre part, les connaissances permettant d'agir sur celles du domaine. Cette caractérisation a été identifiée, spécifiquement pour la conception, par Hatchuel et Weil qui proposent dans leur théorie C-K, voir [Hatchuel and Weil, 2002], la distinction entre les espaces des connaissances et celui des concepts, qui contient des ensembles se déduisant les uns des autres par partitions ou inclusions successives.

Arciszewski constate qu'un changement de paradigme s'opère en ingénierie, d'un paradigme analytique vers un paradigme de connaissances ([Arciszewski et al., 1995]). L'ingénierie était préalablement centrée sur la construction de modèles analytiques des systèmes techniques, afin de développer la compréhension de leur comportement et de produire des connaissances à leur sujet. Maintenant, les connaissances suffisantes sont largement disponibles. L'attention de la conception est portée sur la façon d'utiliser cette connaissance par les moyens des technologies d'information.

A partir de cette distinction, se pose la question de savoir si les connaissances du domaine et celles méthodologiques, procédurales, ... qui permettent d'utiliser les connaissances du domaine sont indépendantes ou non ; les avis sont ici divergents. Toutefois un consensus se forme, en effet les travaux sur les systèmes Common-KADS [Valente et al., 1998], MACAO-II [Matta, 1995], DSTM [Trichet, 1998] ou SYRCLAD [Guin, 1998] aboutissent aux conclusions :

- des tâches génériques de résolution de problèmes peuvent être identifiées indépendamment du domaine ;
- le mode de représentation des connaissances du domaine dépend de la tâche à réaliser. En effet, « *des tâches différentes ou des méthodes différentes de résolution d'un même problème peuvent induire des descriptions différentes qui constituent autant de vues finalisées sur un domaine.* » [Caplat, 2002]

Notons enfin que des méthodes de résolution de problèmes spécifiques peuvent être identifiées en fonction de la tâche à réaliser. Motta note que les méthodes indépendantes des tâches abordent les problèmes à un niveau d'abstraction plus élevé [Motta and Zdrahal, 1998].

III.1.1.2. Lien domaine – démarche de résolution

Dieng, dans [Dieng, 1990], insiste sur le lien entre classes de problèmes et processus d'acquisition des connaissances, précisant l'importance de la bonne compréhension des tâches d'un expert pour y adapter au mieux le mode d'acquisition des connaissances. Si l'on reprend la classification de Clancey ([Clancey, 1985]), qui distingue les problèmes selon qu'ils réfèrent à de l'analyse ou à de la synthèse, la conception de systèmes techniques appartient à la seconde catégorie. La synthèse est ensuite décomposée en spécification, conception et assemblage, comme illustré à la figure III.1..

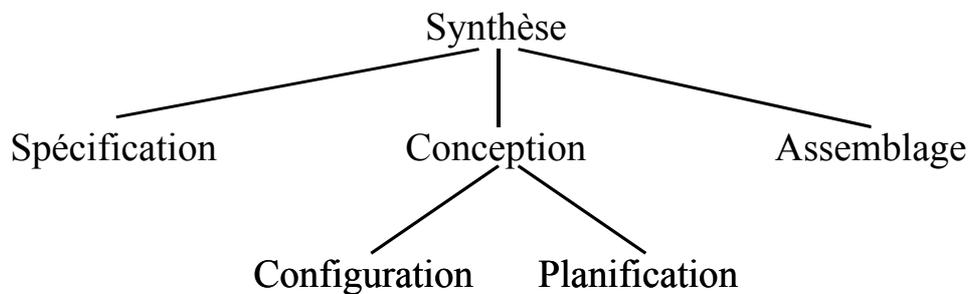


Figure III.1. La classification des problèmes, selon [Clancey, 1985]

Sur la base de cette classification, une taxonomie de tâches génériques a été définie dans KADS. Nous présentons ici les deux premiers niveaux de cette classification.

analyse_de_système

identification // identification d'un état d'un système ou d'une propriété de ce système

prédiction // identification d'un état passé ou futur du système

modification_de_système

réparation // remplacement d'un composant fautif ou suspect

remède // initialisation et contrôle d'un processus neutralisant ou contrebalançant un processus ou un état du système

contrôle // maintien de l'intégrité d'un système dynamique

synthèse_de_système

transformation // modification des composants et de la structure d'un produit

conception // spécification des composants et de la structure d'un produit

planification // combinaison dans une séquence de temps d'actions et d'opérations

modélisation // satisfaction de certains critères de comportement du produit, qui sont dérivés de l'objet à modéliser

Cette description des tâches génériques tend à montrer qu'il est possible de construire des démarches satisfaisant ces tâches, indépendamment du domaine d'application. Cette tendance est appuyée par d'autres auteurs (Mathieu, 1986 #207; Hoc, 1986 #208). En revanche, les démarches de réalisation des tâches génériques mettent en jeu des typologies de connaissances différentes. Si les tâches sont les objectifs de la résolution de problème, les démarches sont les modes d'inférence réalisant ces objectifs. Caplat définit le lien entre les connaissances et les usages par le mode de représentation ([Caplat, 2002]) :

« La nature d'une connaissance en conditionne l'usage et chaque usage trouve dans un paradigme de représentation une forme plus ou moins adaptée. »

Deux critères à prendre en considération lors de la constitution d'un système à base de connaissances sont ici relevés :

- Le lien entre la nature des connaissances et l'inférence traitant ces connaissances. Les inférences étant représentatives des activités de l'expert, leur explicitation doit être conditionnée par les connaissances du domaine.
- l'importance du mode de représentation des connaissances.

Concernant le lien entre la nature des connaissances et le type d'inférence, ou la stratégie de résolution, nous pouvons reprendre la typologie de Hoc, complétée par Caplat.

- Les transformations d'états : le processus de résolution est un cheminement dans un espace d'état ; le « General Problem Solvateur » de Newell et Simon repose sur ce principe. Ce cheminement s'appuie sur les stratégies d'essais erreurs et sur l'analyse moyens-fins (basée sur la comparaison entre l'état existant et l'état souhaité, elle nécessite de pouvoir évaluer l'écart au but). Les connaissances sont représentées sous forme de vecteurs d'état.
- L'induction de structures repose sur la recherche de relations dans un ensemble d'éléments. La stratégie utilisée est alors le test d'hypothèses de solutions potentielles, à partir de données identifiées et débouche sur l'identification de modèles de solutions. Les compléments de séries (1A 2B 4D 8M...) sont un exemple d'induction de structures.
- Dans les problèmes de conception la stratégie utilisée est celle d'un affinement progressif. Le résultat est une représentation détaillée du but sous la forme de contraintes à satisfaire.
- L'analogie consiste à établir des liens entre le problème courant et des problèmes résolus antérieurement.

Il paraît évident que le domaine influe peu pour une stratégie de résolution donnée. En revanche, il apparaît dans la classification précédente qu'il existe des stratégies préférentielles en fonction de la nature des problèmes à résoudre, et, que les modes de représentation des connaissances d'un domaine doivent être adaptés à la stratégie choisie. La démarche, correspondant à la constitution d'une base de connaissances représentant un domaine, en vue de la résolution de problèmes, est illustrée à la figure III.2..

Cette démarche illustre que, pour un domaine donné, le mode de constitution de la base de connaissances dépend des problèmes à y résoudre. Ainsi, de nombreuses bases de connaissances sont adaptées à l'ingénierie et y résolvent des problèmes de natures diverses : coopération, partage d'informations, conception routinière, re-conception...

Mais quel serait le mode adapté à la conception inventive ? Pourquoi les modes de représentation existants ne s'adaptent-ils pas à la conception inventive ? Le prochain paragraphe s'attache à répondre à ces questions.

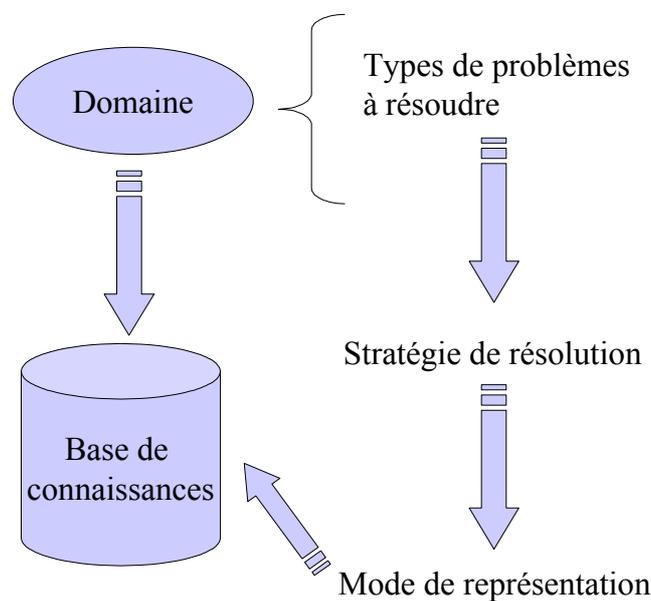


Figure III.2. Démarche de constitution d'une base de connaissances

III.1.2. La représentation des connaissances pour la conception inventive

Nous avons identifié deux types de connaissances à représenter pour la modélisation des processus de conception inventive :

- Les connaissances stratégiques permettant d'agir sur les connaissances du domaine ;
- Les connaissances liées à un domaine.

III.1.2.1. Les connaissances stratégiques

Il a été établi, voir figure III.2., que le domaine oriente la définition des problèmes à résoudre et que c'est à partir de la nature de ces problèmes qu'une stratégie de résolution peut être établie. En conception, plusieurs approches peuvent permettre de définir le domaine d'étude : soit le domaine est limité à celui d'apparition du problème, soit il est étendu à l'ensemble des connaissances d'ingénierie. Le premier cas restreint la recherche de solution, car elle doit être trouvée dans le domaine d'apparition du problème, et, donc, doit orienter vers une conception routinière ; en revanche, l'extension du domaine considéré permet d'ouvrir la recherche de solution et favorise la conception inventive. Plaçons notre problématique sur la conception inventive, le domaine considéré étant celui des connaissances d'ingénierie, indépendamment de tout domaine d'expertise.

Dans le cadre de la conception inventive, les problèmes considérés relèvent de la synthèse, cf. §III.1.2. Celle-ci est confrontée à un certain nombre de limitations, que Simon définit ainsi :

« Notre rationalité est limitée de trois manières. Nous ne connaissons qu'une infime partie des choses que nous aurions besoin de connaître, de celles qui sont nécessaires à l'obtention d'une conception optimale. Et notre pouvoir compilatoire ne nous autorise à compiler qu'une petite partie des implications infinies des choses que nous connaissons. Mais notre rationalité est également limitée d'une troisième manière. Nous stockons ce que nous connaissons dans une portion encyclopédique de notre cerveau habituellement appelée "mémoire à long-terme".

La mémoire à long-terme est indexée et on y accède par un processus appelé reconnaissance. Des stimuli de l'environnement extérieur - un mot sur une page, un dessin, un objet - nous donnent accès à des informations déjà stockées dans la mémoire.

Nous avons une quantité d'informations disponible, mais seule une infime partie peut être portée à notre attention à un moment donné. » [Simon, 1987]

Par ces remarques sur les limitations de l'homme, Simon pose la question de la gestion des connaissances à prendre en compte lors de la conception. La difficulté à trouver des solutions à un problème résulte, bien souvent, d'une quantité importante de connaissances, auxquelles il est difficile d'accéder et organiser, plutôt que d'un manque de connaissances. Un système d'assistance à la conception doit donc accompagner le concepteur afin de filtrer les connaissances à intégrer dans le processus de conceptualisation des connaissances. Mathieu, dans [Mathieu, 1986], définit, plus généralement, la résolution de problèmes comme une suite d'inférences, ayant pour but de « *contrôler quelles connaissances doivent être utilisées pour atteindre un but, et dans quel ordre elles doivent être activées.* »

Le processus de reconnaissance, qui permet de faire appel à la mémoire long terme, nécessite de faire appel à des connaissances déjà acquises. Pour Bonnardel ([Bonnardel, 2000]) les concepteurs « *peuvent évoquer des objets connus ou 'sources', afin de mieux comprendre l'objet à concevoir ; en même temps, ils transfèrent certaines propriétés des sources au problème considéré (la cible).* » L'intérêt de ce transfert pour les problèmes de conception, qui sont à espace non limité, est de pouvoir se référer à des sources d'inspiration très variées. Les concepteurs peuvent ainsi plus facilement évoquer des sources qui leur sont familières, même si ces sources n'ont pas de lien direct avec le problème.

Les connaissances stratégiques identifiées sont celles réalisant :

- des tâches de filtrage des connaissances ;
- des tâches de construction d'une représentation permettant la réutilisation de connaissances antérieures.

Le processus de filtrage correspond à la formulation du problème, dont l'importance est relevée par Maier dans [Maier, 1963]. Il montre que la résolution d'un problème dépend de l'identification d'un obstacle à surmonter, et que la nature de la solution varie en fonction du mode de représentation de cet obstacle. L'identification d'un obstacle qui soit surmontable, et représentatif du problème, est une étape clé de la résolution du problème.

Cette formulation du problème, en tant qu'identification d'un obstacle qui soit surmontable, doit se faire, tout en permettant la reconnaissance d'une situation de conception antérieure, facilitant ainsi l'élimination de l'obstacle en question. Ces antériorités, encore appelées précédents de conception dans [Oxman, 1994], sont des éléments distincts de connaissance, rendus uniques par des caractéristiques conceptuelles. Les précédents de

conception sont les explications des apports pertinents des conceptions antérieures, ainsi que les liens appropriés entre les différents précédents.

Si l'on compare les apports de la TRIZ à la conception, par rapport à ces connaissances stratégiques, on constate que les éléments de base y sont présents, à savoir :

- un guide méthodologique de capitalisation des connaissances utiles à la résolution du problème, ARIZ ;
- l'utilisation de l'analogie comme mode de résolution des problèmes par le recours à des situations antérieures génériques.

Notre proposition d'amélioration du formalisme de la TRIZ, pour la constitution d'un outil d'aide à la formulation de problème, se doit de :

- capitaliser les connaissances utiles à la résolution du problème par l'identification d'un obstacle à surmonter ;
- identifier un obstacle qui soit surmontable, afin de s'assurer de la solvabilité du problème ;
- orienter vers des modes de résolution issus de situations précédemment résolues.

Le recours aux outils de la TRIZ permet de répondre conjointement aux deux dernières contraintes. La formulation, selon un modèle de problème issu de la TRIZ, assure l'existence d'une méthode générique de transformation du modèle qui, par affinement progressif, mène à l'identification d'une solution. La genèse des outils de résolution de la théorie s'appuie sur des précédents de conception, précédemment résolus et caractérisés par des éléments génériques discriminants.

III.1.2.2. Les connaissances du domaine

Le domaine relatif à la conception inventive est large, il concerne en effet toute l'ingénierie. Dès lors, la modélisation des connaissances du domaine devient un problème particulièrement ardu. Ceci est dû à plusieurs raisons :

- L'étendue des connaissances à capitaliser nécessiterait quasiment de construire une représentation du monde ; or, une bonne représentation se heurte à l'élicitation des connaissances expertes [Champin, 2002].
- La multiplicité des points de vue, car, l'expert possédant les connaissances d'un domaine n'étant pas celui qui aura à recourir à ces connaissances modélisées, l'interprétation ne peut donc en être efficiente [Rechenmann, 1995]. D'autant que les modèles de connaissances d'un domaine donné peuvent évoluer au cours du temps pour un même individu [Richards and Simoff, 2001].
- Le mode d'accès aux connaissances n'est pas connu a priori. Nous revenons ici à la notion de reconnaissance d'objets. Le recours aux connaissances se fera en fonction du problème à résoudre ; pour deux problèmes distincts le recours à une même connaissance peut se faire de deux manières différentes. Le mode de représentation des connaissances doit donc être souple et ouvert.

La constitution d'une base de connaissances pour la conception inventive n'est pas un objectif réaliste.

De nombreuses approches proposent des systèmes d'assistance à la conception par la constitution de démarches de réutilisation de cas [Ball et al., 2001; Champin, 2002; Gzara et al., 2003; Kolodner, 1993; Oxman, 1994]. Ces méthodes apportent, comme intérêt principal, la possibilité de contextualiser les connaissances. En particulier, peuvent être incluses les connaissances concernant le « *raisonnement ayant mené du problème à la solution* » [Fuchs and Mille, 2000] qui fournissent un contexte pour comprendre et évaluer une situation [Kolodner, 1993]. Cette capitalisation, en vue de la réutilisation de cas, n'est pas intégrée à la TRIZ ; elle est recommandée dans le déroulement d'ARIZ, mais son application est orientée, de manière à améliorer l'efficacité de la méthode générale, et non, dans le but de permettre une mise en œuvre accélérée par le recours aux situations antérieures. Certes le mode de résolution, une fois le modèle de problème construit repose sur un raisonnement par analogie, mais la construction du modèle de problème et la spécification du modèle de solution pourraient s'appuyer sur des cas, à un niveau de généralisation moindre.

Se pose la question du mode de représentation des cas ; plusieurs objectifs contraignent un potentiel cadre de capitalisation :

- le cas capitalisé doit être représentatif de la démarche de formulation du problème et de sa résolution ;
- le modèle doit intégrer les connaissances permettant de contextualiser le cas et pas uniquement le problème et sa solution ;
- la capitalisation du cas doit se faire au cours de la résolution du problème, afin de ne pas nécessiter un surcroît de travail ;
- la capitalisation des cas doit permettre de construire progressivement une image contextualisée des domaines de l'ingénierie.

Nous proposons d'apporter des réponses, au moins partielles, aux objectifs précités, en constituant un modèle de formulation des problèmes selon les cadres de la TRIZ, réponses qui permettent à la fois, d'accompagner de faciliter cette formulation et de capitaliser les cas résolus.

III.1.2.3. Les connaissances de la TRIZ à modéliser

Notre objectif de formalisation passe avant tout par l'identification des concepts propres à la formulation des problèmes selon les cadres de la TRIZ. Pour cela il nous faut comprendre la nature des informations qui sont incluses dans une formulation complète du problème. Souhaitant permettre un recours aisé aux règles de transformation des modèles, il nous faut également identifier les informations à collecter pour la mise en œuvre de ces règles, dans le cas où elles seraient différentes des premières.

Cette démarche d'analyse, à la fois des cadres de formulation existants et des règles de transformation permet de vérifier, et le cas échéant de construire, la cohérence de la théorie.

Lesdits concepts sont les briques de base des connaissances stratégiques de représentation des problèmes, selon les cadres de la TRIZ. Nous pouvons considérer cet ensemble de concepts comme l'ontologie de la représentation des problèmes, selon les cadres de la TRIZ. Cette ontologie est la base de la capitalisation des connaissances domaine. En effet, étant une ontologie de description, elle décrit la manière dont toute information doit être représentée pour être en cohérence avec la théorie.

La constitution d'une base de connaissances domaine peut se faire ainsi par une simple capitalisation des cas selon cette ontologie.

III.1.3. Démarche de modélisation

III.1.3.1. Choix du corpus pour la constitution du modèle

Les objectifs de la modélisation ayant été fixés, nous allons décrire la démarche de modélisation que nous avons suivie.

Comme déjà explicité au chapitre II, le corpus principal, dans lequel nous identifions les concepts de formulation des problèmes de la TRIZ, est l'algorithme ARIZ (cf. Annexes-II, III et IV). Celui-ci mène à la construction de plusieurs modèles de problèmes successifs :

- la définition de contradictions techniques ;
- la définition d'une contradiction physique ;
- la définition d'un résultat idéal, représentation utopique de l'objectif du problème.

Ce sont principalement les trois premières parties de l'algorithme qui participent à la formulation du problème, les parties suivantes étant orientées vers les recherches de solutions et la remise en cause du problème en cas de non-résolution.

L'analyse de ce corpus permet d'identifier un certain nombre de concepts et de liens entre ces concepts, mais pas toujours d'en expliciter le sens. Cette précision doit être obtenue par l'analyse de la littérature sur la TRIZ, et par l'observation de la mise en œuvre par les experts en situation de résolution de problèmes.

Un second ensemble de textes a été très utile pour la construction du modèle : les standards de résolution (cf. Annexe-I). Cet ensemble de soixante-seize règles heuristiques de transformation de modèle est une richesse descriptive des situations problématiques pouvant être résolues par la TRIZ. L'analyse du caractère discriminant de chaque règle permet d'établir l'ensemble des critères de distinction des situations problématiques, et donc, de proposer un modèle de formulation complet, orientant directement vers une règle adéquate de transformation du modèle.

Les concepts identifiés par l'analyse de ces corpus ont été modélisés dans Protege-2000 [Informatics, 2003]. Protege-2000 est un environnement logiciel générique et extensible qui permet aux utilisateurs de modéliser, de solliciter et d'exploiter des ontologies en tout domaine. Plus spécifiquement, Protege-2000 fournit un éditeur de modélisation d'ontologies, avec lesquelles les experts du domaine peuvent représenter leurs connaissances. Protege-2000 adopte

une vue basée sur les « frames »⁷, que nous appellerons concepts dans la suite, pour les ontologies. En conséquence, un ensemble de concepts est organisé en une hiérarchie pour la relation de subsomption, pour représenter les concepts dans le domaine ciblé, et des attributs leur sont rattachés pour représenter leurs propriétés. Les valeurs pouvant être prises par les slots, nous utiliserons dorénavant le terme attribut, sont restreintes par des facettes, comme la cardinalité, le type et la plage. Les classes sont des modèles pour des instances individuelles, ayant des valeurs particulières d'attributs [Crubézy and Musen, 2004].

Cette représentation centrée-objet des concepts permet de visualiser aisément les liens entre les objets représentés, et il y est possible d'affecter une instance d'un concept particulier comme valeur d'attribut d'un autre concept. Ceci nous est nécessaire pour représenter les liens entre les différents éléments de modèles de problème.

III.1.3.2. Exploitation du modèle

Le modèle établi, son exploitation se fait par la définition de questions guidant l'instanciation de celui-ci. L'appel à ces questions doit se faire selon une stratégie d'instanciation. La stratégie que nous avons établie se base sur les différents points d'entrée de la formulation des problèmes ; cela signifie que nous avons établi une liste générique de situations problématiques pour lesquelles nous pré-supposons l'existence de connaissances. Ainsi, si l'objectif est d'améliorer la réalisation d'une fonction du système, nous commençons par renseigner le système et la fonction à améliorer.

La mise en œuvre de ces stratégies, et l'exploitation du modèle en général se font par l'écriture de règles dans Jess [Friedman-Hill, 2002]. Jess est un moteur de système expert basé sur des règles, permettant de concevoir des applets Java et des applications utilisant des

⁷ Protege-2000 utilise un langage de représentation selon le schéma : Classes, Instances, Slots, Facettes [Gennari et al., 2003] Gennari, J. H., Musen, M. A., Ferguson, R. W., Grosso, W. E., Crubezy, M., Eriksson, H., Noy, N. F. and Tu, S. W. (2003) The evolution of Protege: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, pp. 89-123.. Les classes regroupent des ensembles d'objet dont chaque élément particulier spécifié est une instance. Les slots représentent les attributs discriminants de ces entités, hérités au sein d'une même classe et distinguant les classes. Les contraintes sur les classes et sur les slots (attributs) sont définies par les facettes [Tu et al., 1995] Tu, S. W., Eriksson, H., Gennari, J. H., Shahar, Y. and Musen, M. A. (1995) Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: application of PROTEGE-II to protocol-based decision support. *Artificial Intelligence in Medicine*, 7, pp. 257-289.

connaissances, fournies sous forme de règles déclaratives et de faits [Friedman-Hill, 2003]. Le lien entre Jess et Protege est fait grâce à JessTab [Eriksson, 2003a]. JessTab fournit une fenêtre de console Jess, comme onglet dans Protege, ainsi qu'un ensemble d'extensions permettant d'avoir accès, dans Jess, aux bases de connaissance de Protege, et ainsi, d'exploiter les bases de connaissances de Protege [Eriksson, 2003b].

Une interface graphique très simple a été développée en Java à partir de Jess afin de collecter les connaissances d'instanciation du modèle de problème.

III.2. un modèle de formulation des problèmes selon les cadres de la TRIZ

La modélisation construite et présentée ici s'attache à :

- Permettre la construction d'une formulation de problèmes selon les cadres de la TRIZ, afin d'identifier clairement le cœur du problème à résoudre et d'être en mesure de recourir aux outils de la TRIZ pour faire évoluer la formulation rapidement vers une solution.
- Clarifier les notions nécessaires à une bonne formulation des problèmes selon les cadres de la TRIZ.
- Proposer une sémantique qui soit complète et cohérente et supprime toutes les ambiguïtés présentes dans les textes sur la TRIZ.
- Expliciter le mode de transition entre les cadres de formulation de problèmes de la TRIZ.

La démarche de construction de l'ontologie est composée de cinq étapes :

- identification des concepts à représenter ;
- représentation et hiérarchisation des concepts ;
- validation du modèle ;
- inférence sur les connaissances du modèle ;
- création d'une interface d'instanciation.

Ces étapes s'appuient sur des principes scientifiques et sont réalisées par l'exploitation de logiciels, détaillés dans le tableau IV.1.

Ce chapitre est consacré à la description de ces étapes, permettant ainsi une bonne compréhension des concepts manipulés et du formalisme employé.

Lors de ce chapitre, nous allons détailler le recours aux différents outils participant à la constitution du modèle et à son opérabilité. L'identification des concepts à représenter a été réalisée en analysant sémantiquement les textes grâce à LIKES, ces concepts ont ensuite été représentés et hiérarchisés en représentation centrée-objet. Cette opération s'est appuyée sur le logiciel Protege-2000. Le modèle constitué a été validé par une modélisation en logique de description, programmée et évaluée dans CICLOP. Enfin la définition de règles dans JESS pour

exploiter le modèle de Protege-2000, via l'interface JessTab, et la définition d'une interface en JAVA permette d'instancier le modèle, pour assister la formulation des problèmes.

Objectif	Concept scientifique	Outil informatique
Identifier les concepts à représenter	Analyse automatique des textes	LIKES
Représenter et hiérarchiser les concepts	Représentation centrée-objet	Protege-2000
Valider la cohérence du modèle	Logiques de description	CICLOP
Inférer sur les connaissances du modèle	Systèmes à base de règles	JESS
Exploiter le modèle	"Interface utilisateur"	JessTab et Java

Tableau IV.1. Concepts et outils de construction du modèle

III.2.1. Identification des concepts

III.2.1.1. *LIKES, outil de Traitement Automatique Linguistique*

LIKES [Rousselot, 2002] est une station d'ingénierie linguistique destinée à traiter des corpus constitués d'un ou plusieurs textes. Différents traitements sont disponibles, afin de fournir des données interprétables par l'utilisateur, sous la forme de listes de séquences de mots, candidates à avoir une interprétation particulière, par exemple : termes, relations ou ensembles de termes à interpréter comme une classe générique.

Des outils d'analyse du corpus permettent :

- d'établir la fréquence d'utilisation des termes, identifiant ainsi rapidement les concepts les plus souvent répétés et à définir ;
- d'identifier les segments répétés. Cette tâche a pour but d'extraire les groupes nominaux, candidats à la définition en tant que termes du corpus. Les termes sont les éléments significatifs d'un domaine. Ils sont utilisés pour exprimer les concepts et objets principaux d'un domaine ;
- de faire la synthèse de schémas exprimant des relations sémantiques découvertes dans le corpus, et de les exprimer de différentes manières. Les schémas sont des cadres morpho-syntaxiques que le logiciel synthétise et demande à l'utilisateur d'éditer. Le résultat est une expression validée qui est finalement utilisée dans les textes ;
- de générer des tâches terminologiques autorisant la définition de termes, de relations entre ceux-ci (synonyme, antonyme, associé à) et de concepts auxquels peuvent être reliés les termes.

Cet outil permet donc d'aider à identifier l'ensemble des notions présentes dans les textes étudiés. Nous allons décrire maintenant les concepts que nous avons retenus pour notre modélisation dans les deux textes étudiés : ARIZ et les standards de résolution de problèmes.

III.2.1.2. Les concepts issus de l'algorithme ARIZ

Nous présentons ici l'analyse détaillée de l'algorithme de résolution des problèmes inventifs, dans sa version de 1985. Cette version est la plus répandue, car la dernière à avoir été développée par Altshuller. Afin de ne pas alourdir l'exposé, l'analyse que nous présentons porte sur les trois premières parties de l'algorithme, celles participant pleinement à la formulation du problème, et seules, les étapes définissant des concepts de formulation importants, sont précisées.

Partie 1. Analyse du problème

Etape 1.1. Formulation du mini-problème

C'est certainement l'une des étapes les plus critiquées, sans doute à juste titre, de cette version d'ARIZ. Ce stade initial suppose que le problème soit déjà clairement identifié et qu'ARIZ ne se veuille qu'un outil d'accompagnement de la modélisation du problème. Cependant, les indications laissées par Altshuller nous montrent qu'il doit s'agir d'une réelle première étape dans l'analyse d'un problème. Toutefois, en comparant les versions antérieures d'ARIZ (cette partie est décomposée en 6 étapes dans ARIZ-71 [Altshuller, 1973] et en 9 étapes dans ARIZ-77 [Arciszewski, 1988]), il est légitime de se demander s'il n'y a pas eu un appauvrissement de cette étape. De fait, certains experts reconnaissent qu'il est plus facile de se former à l'approche de la résolution des problèmes par TRIZ à travers la version ARIZ-77 [Savransky, 2000].

Les objectifs sont les suivants:

- Obtenir une description sommaire de la **fonction du système**
- Etablir une première liste des **composants du système**
- Etablir une première liste des **ressources du système**
- Une première formulation de **contradiction technique**
- Une première formulation du **résultat idéal final**

Pour être efficiente, cette étape doit être accompagnée d'une démarche de déblocage de l'inertie psychologique, par l'utilisation de termes génériques, se situant à un niveau abstrait, afin de dresser un modèle du problème explicite et non ambigu.

Les objectifs ci-dessus sont respectés par l'utilisation du patron suivant:

Un système technique pour <décrire la fonction du système> inclut <listes les principales parties et ressources du système>.

Contradictions Techniques :

Si <A>, alors est positif et <C> est négatif.

Si <A> est opposé, alors < C > devient positif mais < B > devient négatif.

Etape 1.2. Définition de la paire en conflit

Cette partie a pour but l'identification de la paire: **Produit / Outil**.

Le produit est l'élément qui nécessite un traitement (fabrication, déplacement, modification, amélioration, ...). Il représente l'objet sur lequel porte la fonction.

L'outil est l'élément qui agit directement sur le produit.

Etape 1.5. Intensification⁸ du conflit

Il faut indiquer l'état extrême du système ; pour y parvenir, il faut :

- Identifier le paramètre influençant l'effet néfaste.
- Déterminer le sens à donner à cette variation dans le but d'augmenter l'effet néfaste.
- Augmenter progressivement le paramètre identifié, dans le sens déterminé, jusqu'à obtenir l'effet néfaste maximal.

Partie 2 : analyse du modèle de problème

Etape 2.1. Définition de la zone opératoire

Cette description se fait en trois étapes:

- Analyse de la zone géographique où l'effet néfaste apparaît.
- Analyse de la zone géographique où l'effet positif devrait être.

⁸ L'intensification est une technique très utilisée dans la TRIZ pour lutter contre l'inertie psychologique. Elle repose sur la définition de conditions extrêmes, par la modulation de valeurs de paramètres de la situation.

- Comparaison des deux zones ; si celles-ci n'ont pas d'aire commune, la résolution du problème peut se faire par séparation spatiale au macro-niveau ; si, au contraire, elles sont totalement identiques, il faut considérer les opportunités de résoudre le problème dans l'espace, au micro-niveau.

Etape 2.2. Définition du temps opératoire

Cette description se fait en trois étapes:

- Analyse de la période temporelle durant laquelle l'effet néfaste apparaît.
- Analyse de la période temporelle durant laquelle l'effet positif devrait être.
- Comparaison des deux périodes ; si celles-ci n'ont pas de durée commune, la résolution du problème peut se faire par séparation temporelle au macro-niveau ; si, au contraire, elles sont totalement identiques, il faut considérer les opportunités de résoudre le problème dans le temps, au micro-niveau.

Etape 2.3. Définition des ressources de substances et champs

On cherche ici à définir **les ressources** en termes de **substances** et de **champs** pour :

- Le système : outil et produit.
- L'environnement : systèmes adjacents, ressources naturelles (gravité, par ex.).
- Le super système : matériaux gaspillés ou ressources non onéreuses disponibles.

Une classification des ressources peut être faite sur la base de l'occurrence dans différents problèmes répertoriés.

Partie 3 : Définition du Résultat Idéal Final et de la contradiction physique

Etape 3.1. Formulation du Résultat Idéal Final 1

Sur la base du patron proposé par l'algorithme, il faut clarifier le rôle à jouer par l'élément-solution, dans la ligne d'évolution **d'accroissement de l'idéal**.

Etape 3.2. Intensification du Résultat Idéal Final 1

L'intensification passe par **l'interdiction d'utiliser une ressource externe au système**. L'élément X doit être l'une des ressources (ou une combinaison, une altération) listées à l'étape 2.3.

Etape 3.3. Identification de la contradiction physique au macro-niveau et Etape 3.4. Identification de la contradiction physique au micro-niveau

La définition de la contradiction physique s'appuie sur les analyses effectuées dans les étapes précédentes, mais est enrichie par un questionnement sur l'utilisation systématique des ressources connues (celles présentes et listées auparavant).

Les concepts identifiés par l'analyse d'ARIZ sont de différentes natures, dont certains participent à la modélisation du monde réel.

- La notion de fonction, rattachée à un système, est la fonction principale pour laquelle le système est conçu ;
- Des composants du système qui est un assemblage d'éléments permettant de réaliser la fonction principale ;
- Les ressources du système sont les éléments qui n'entrent pas directement dans la réalisation de la fonction principale, mais sont disponibles à proximité du système et pourront donc être utilisés, au moment de la résolution du problème.

D'autres concepts participent à la description du rôle des éléments dans le problème.

- Le produit est l'élément sur lequel est réalisée une fonction, étudiée dans le problème. Cette fonction peut être à améliorer, à réaliser ou encore à faire disparaître.
- L'outil est l'élément agissant directement sur le produit pour réaliser une fonction.

Enfin, certains concepts sont liés à la formulation du problème.

- La notion de contradiction est un mode de représentation du problème.
- L'effet utile est l'effet à assurer lors de la résolution du problème.
- L'effet néfaste, apparaissant lorsqu'on réalise l'effet utile, est à l'origine de la contradiction.

- La zone et le temps opératoires permettent d'intégrer, aux données du problème, les spécifications de l'espace géographique et de moment d'apparition du problème.
- Les notions d'idéalité et de résultat idéal final participent à la clarification des tendances d'évolution à suivre lors de la résolution du problème.

III.2.1.3. Les concepts issus des standards de résolution

Le texte des standards est organisé en cinq classes, elles-mêmes découpées en sous-classes, représentant le degré d'évolution du modèle substance-champ considéré, et proposant des règles de modification, adaptées à ce degré d'évolution. Une classe, a fortiori une sous-classe, réfèrent à une typologie de problème, et, de fait, sous-entendent certaines caractéristiques du problème. Ainsi, des connaissances de catégorisation des problèmes sont incluses dans la définition des classes et non rappelées spécifiquement dans chaque standard de la sous-classe.

Par exemple, la première sous-classe, notée 1.1, réfère à la synthèse des modèles substance-champ, c'est-à-dire que les transformations proposées par les standards de cette sous-classe concernent des modèles incomplets. Pourtant, le premier standard de cette sous-classe, noté 1.1.1, ne rappelle pas cette nécessaire condition d'application :

« Si un objet résiste aux changements nécessaires et si les conditions du problème ne contiennent pas de contraintes à l'introduction de substances et de champs, alors on résout le problème par la synthèse du vèpole en introduisant les éléments qui manquent. »

Un modèle substance-champ complet est défini comme composé de deux substances et d'un champ d'interaction entre les substances. Pourtant, la considération du standard 1.1.6, appartenant également à la sous-classe de synthèse des modèles substance-champ, remet cette définition en cause :

« Si l'on a besoin du régime minimal (dosé, optimal) d'action et si, selon les conditions du problème, il est difficile ou impossible de l'assurer, alors il faut utiliser le régime maximal et enlever l'excès. Dans ce cas, on enlève l'excès de champ par la substance et l'excès de substance par le champ. »

En effet, dans le standard ci-dessus, le modèle substance-champ initial est composé de deux substances et d'un champ d'interaction, mais ce champ est appliqué de manière inadaptée à un bon fonctionnement. L'explication, sur l'appartenance de ce standard à la sous-classe de synthèse des modèles substance-champ, est que la substance qui génère l'excès de champ n'a pas

à être introduite dans le modèle de problème. Ainsi, seuls sont considérés une substance et un champ, et le modèle est alors effectivement incomplet. Mais cette considération n'est pas satisfaisante car les règles d'inclusion des éléments dans le modèle de problème n'existent pas. Comment dès lors savoir si un tel élément, pourtant générateur d'un effet néfaste, doit être inclus ou non ? Pour lever cette ambiguïté, nous distinguons modèle incomplet et modèle fonctionnel. Un modèle non fonctionnel peut être complet, de par la cardinalité de ces éléments, mais ne pas être satisfaisant ; la sous-classe 1.1 concerne alors la synthèse des modèles non fonctionnels, de manière plus générale.

Les concepts identifiés par l'analyse des textes des standards sont :

- le modèle substance-champ, modèle de représentation des interactions entre les éléments composants le système, notamment ceux impliqués dans le problème ;
- les champs et les substances caractérisant les éléments par leur nature.

D'autres concepts participent spécifiquement à la caractérisation du problème, à savoir :

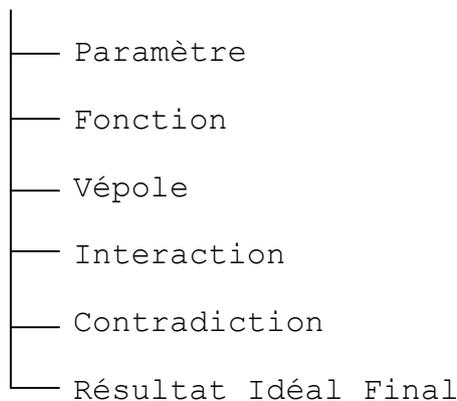
- la qualification des interactions entre les éléments : insuffisante, satisfaisante, excessive ou néfaste ;
- la complétude des modèles substance-champ ;
- la fonctionnalité des modèles substance-champ ;
- la possibilité d'ajouter des substances, des additifs ou des champs au système considéré ;
- la nécessité de préserver le contact entre des éléments.

Les concepts ci-dessus, ajoutés à ceux identifiés dans ARIZ, permettent la constitution d'un modèle de représentation des problèmes selon les cadres de la TRIZ.

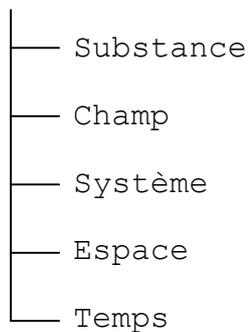
III.2.1.4. Liste des concepts à modéliser

Nous présentons ici les concepts que nous souhaitons intégrer dans notre étude, car participants à la formulation des problèmes selon les cadres de la contradiction physique et du modèle substance-champ. La hiérarchisation proposée repose, d'une part, sur les concepts qui représentent des entités réelles, comme les systèmes, les champs. Ces entités sont la base de la constitution d'un existant domaine, une entreprise peut représenter toutes ces ressources selon ces concepts. Nous regroupons, d'autre part, les concepts inhérents aux modèles qui sont propres à la formulation de problèmes, et, par conséquent, spécifiques à une étude. Nous retrouvons dans cette classe, entre autres, les modèles de la contradiction et substance-champ.

Modèle



Ressources



Nous allons, dans le chapitre suivant, décrire précisément la construction du modèle représentant l'ensemble de ces concepts.

III.2.2. Construction du modèle de représentation des problèmes

Les concepts à représenter étant identifiés, commence le travail de modélisation. Nous présentons ici la constitution du modèle en représentation centrée-objet, support du logiciel d'aide à la formulation de problèmes la validation du modèle par les logiques de descriptions et la définition des règles liées aux connaissances du domaine modélisé.

III.2.2.1. Construction du modèle en représentation centrée-objet dans Protege-2000

Ce modèle est la représentation centrée-objet des concepts de formulation des problèmes, selon les cadres de la TRIZ. Tous les concepts de la théorie ne sont donc pas représentés ; en particulier, sont absentes les lois d'évolution des systèmes techniques, ainsi que la vision multi-écrans.

Le but de ce travail étant de faciliter l'implantation des cadres de formulation des problèmes de la TRIZ dans les entreprises, nous intégrons une représentation de la vue fonctionnelle. Cette intégration permet d'explicitier le lien entre les différents points de vue et de mieux rendre compte du positionnement des modèles de la TRIZ.

La description paramétrique des objets, répandue en Intelligence Artificielle et base de l'approche de Khomenko [Khomenko, 2001] sur la généralisation des concepts de la TRIZ à la résolution de problèmes non techniques, est reprise dans le modèle. Dans ce cadre, une fonction peut alors être décrite comme la variation de la valeur d'un paramètre d'un objet, et nous définissons le concept de fonction ainsi :

Fonction

```
[produit = Instance, classes = {Systeme, Substance}
parametre = Instance, classes = {Parametre}
valeur_initiale = String
valeur_finale = String
systeme = Instance, classes = {Systeme}
type = {Utile, Nefaste}
importance = {Principale, Technique}
fonction_mere = Instance, classes = {Fonction}
valeur_objectif = String]
```

La réalisation de la fonction a pour but de modifier un objet de l'environnement, appelé `produit` pour la fonction considérée. La modification de cet objet, de ce produit, se fait par un changement de valeur pour l'un des paramètres de ce produit. Ce paramètre passe de la valeur

valeur_initiale à la valeur valeur_finale. Cette fonction doit être rattachée à un systeme, qui est un artefact permettant la réalisation de cette fonction. L'action de ce système sur le produit permet de réaliser la fonction. La décomposition fonctionnelle se retrouve dans le modèle par l'identification d'une fonction_mere. Enfin, un ensemble d'attributs est défini dans le but de caractériser la fonction par rapport au problème abordé. Ainsi, la fonction peut être de type utile ou nefaste pour une situation spécifique donnée. Si la fonction est considérée néfaste, le problème s'attachera à en supprimer la réalisation. Par ailleurs, l'importance de la fonction dans un système donné est considérée selon le rôle que joue la fonction, en nous inspirant de la description issue de l'analyse fonctionnelle [NF X 50 150].

- La fonction est une raison d'être du système, elle est alors dite principale.
- La fonction participe à la réalisation d'une fonction principale, mais n'est qu'un mode de réalisation de cette fonction principale, elle est alors dite technique.

Enfin, si la fonction est utile et que l'on souhaite en accroître l'efficacité, cet accroissement est caractérisé par l'attribut valeur_objectif qui est la nouvelle valeur_finale que doit avoir le produit après la réalisation de la fonction.

Le concept de systeme est défini selon la première des lois d'évolution des systèmes technique [Salamatov, 1996] qui définit que tout système pour être fonctionnel doit être composé de quatre éléments jouant, chacun, un rôle spécifique :

- le moteur : il adapte l'énergie extérieure en énergie utilisable par le système ;
- l'outil : il agit sur le produit pour lequel la fonction est rendue ;
- la transmission : elle transmet l'énergie à travers le système ;
- le contrôle : il adapte la réalisation de la fonction aux fluctuations de l'environnement.

Dans la représentation centrée-objet, le concept de système est défini comme suit :

Systeme

```
[moteur = Instance, classes = {Systeme, Substance}
transmission = Instance, classes = {Systeme, Substance}
outil = Instance, classes = {Systeme, Substance}
controle = Instance, classes = {Systeme, Substance}
parametre = Instance, classes = {Parametre}
considere = Boolean
```

```
role = String]
```

L'attribut `considere` permet de préciser si le système est un élément impliqué dans le problème considéré, et ce, dans l'optique d'une capitalisation des connaissances et de la constitution d'une base permettant la ré-utilisation. Enfin l'attribut `role` définit, le cas échéant, si le système participe à la réalisation d'une fonction utile ou néfaste, dans le problème considéré.

La contradiction, dans notre modèle, réfère au concept de fonction en le considérant comme la cause des états contradictoires d'un paramètre :

```
Contradiction
```

```
[elements = Instance, classes = {Systeme, Substance}
parametre = Instance, classes = {Parametre}
valeur1 = String
fonction1 = Instance, classes = {Fonction}
temps1 = Instance, classes = {Temps}
espace1 = Instance, classes = {Espace}
valeur2 = String
fonction2 = Instance, classes = {Fonction}
temps2 = Instance, classes = {Temps}
espace2 = Instance, classes = {Espace}]
```

Par rapport à la représentation originale de la contradiction, sont ajoutés des éléments collectés durant le déroulement d'ARIZ mais non inclus dans le modèle de contradiction. Ce sont les notions d'espace et de temps d'apparition du besoin de la valeur pour le paramètre. Ainsi, `valeur1` du parametre de `element` est requise, afin de réaliser `fonction1` durant `temps1` et en la zone `espace1`.

Le modèle substance-champ est lui aussi plus complet que la représentation graphique telle qu'elle existe dans la littérature TRIZ :

```
Vepole
```

```
[elements = Instance, classes = {Champ, Systeme, Substance}
interactions = Instance, classes = {Interaction}
completude = Boolean
fonctionnalite = Boolean
necessite_de_contact = Boolean
ajout_champ = Boolean
```

```
ajout_substance = Boolean  
ajout_additif = Boolean]
```

Nous retrouvons, dans le modèle, les éléments constitutifs du modèle substance-champ qui peuvent être, indépendamment, des champs, des systèmes ou des substances. Des interactions existent entre ces éléments ; elles seront caractérisées par un concept que nous allons détailler plus loin. Enfin, le modèle substance-champ est défini par la valeur des attributs de complétude, de fonctionnalité, la possibilité de rompre le contact entre des éléments, d'ajouter des substances, des additifs ou des champs. De la valeur de ces attributs caractéristiques dépendent les règles de résolution qui seront associées pour modifier le modèle.

Un concept nouveau est introduit dans celui de modèle substance-champ, il s'agit du concept d'interaction :

```
Interaction  
[champ = Instance, classes = {Champ}  
fonction = Instance, classes = {Fonction}  
temps_operationnel = Instance, classes = {Temps}  
espace_operationnel = Instance, classes = {Espace}  
  caractere = Symbol = {Insuffisante, Satisfaisante, Excessive,  
Nefaste}]
```

Ce concept définit la nature des interactions entre les éléments constituant le modèle substance-champ; l'attribut `champ` caractérise la nature du champ et l'attribut `fonction` précise la cause justifiant l'interaction. Les notions d'`espace_operationnel` et de `temps_operationnel` permettent d'affiner la description, en situant spatialement et temporellement la réalisation de la fonction considérée. L'attribut `caractere` détermine la nature de l'interaction, par rapport au problème considéré, et celle-ci peut être jugée :

- `Satisfaisante`, n'impliquant aucune modification à lui apporter ;
- `Insuffisante`, elle est à préserver mais son efficacité doit être renforcée ;
- `Excessive`, elle est à préserver mais son degré d'action doit être limité ;
- `Nefaste`, l'interaction doit être supprimée.

L'ensemble des concepts définis est présenté dans l'annexe V. Nous allons montrer, dans le paragraphe suivant, comment la cohérence de ce modèle a été justifiée par le recours à la logique de description.

III.2.2.2. Validation du modèle en logique de description

La construction d'une représentation de l'ontologie en logique de description a été réalisée [Bultey, 2004] (cf. Annexe-VI). Sur la base de l'ontologie proposée, a été construite une représentation en logique de description des concepts de formulation de problèmes de la TRIZ. Les logiques de description peuvent être considérées comme des logiques dont les constructions principales sont des termes. Elles s'appuient sur la définition de concepts et de rôles, permettant de définir une relation binaire entre concepts. La relation de hiérarchisation des logiques de description est la subsomption : un concept A subsume un concept B si A est plus général que B, c'est-à-dire l'ensemble des individus représenté par A contient l'ensemble des individus représenté par B.

La représentation centrée-objet impose de construire la hiérarchisation. La logique de description calcule la hiérarchie entre les concepts. L'intérêt que nous voyons dans le recours aux logiques de description est, de fait, la validation de l'ontologie car elle permet de tester la cohérence, à la fois de l'ontologie et de sa hiérarchisation. Si un concept irréaliste est introduit, l'incohérence est identifiée par le système ; par exemple, si un modèle substance-champ incomplet est défini fonctionnel, il est incohérent. De même, si un concept est calculé comme un sous-ensemble d'un autre concept non compatible, cela permet de détecter des manques dans la définition des-dits concepts. Par exemple, si le concept *Contradiction* est calculé comme étant un sous-concept de *Temps*, il faut ajouter un paramètre discriminant entre les deux.

La validité de la modélisation est testée dans le raisonneur CICLOP [de Bertrand de Beuvron et al., 1999] développé au LICIA. Son langage est fondé sur la séparation entre les entités et les rôles, correspondant respectivement aux classes et attributs de la représentation centrée-objet. Une terminologie est alors constituée des ensembles déclaratifs de restriction et de définition de concept [Bultey, 2004]. De plus, CICLOP possède un module qui permet de détecter des anomalies potentielles dans la définition des concepts. En principe, les fils d'un concept doivent être disjoints et leur union doit correspondre à l'extension du concept. Si tel n'est pas le cas, CICLOP l'identifie, au choix de l'utilisateur de considérer l'incomplétude de la définition comme représentative de la réalité ou de la compléter.

Les restrictions sont déclarées selon la syntaxe suivante :

- AND permet de définir une conjonction d'expressions conceptuelles ;
- ALL représente la quantification universelle, précise le co-domaine d'un rôle ;

- SOME représente la quantification existentielle, introduit un rôle et affirme l'existence d'au moins un couple d'individus en relation, par l'intermédiaire de ce rôle ;
- NOT correspond à la négation ;
- OR permet de définir la disjonction.

Nous présentons ici deux rôles pour illustrer la syntaxe. Le premier est la définition du rôle parametre qui lie le concept Fonction ou le concept Contradiction au co-concept Parametre :

```
(define-atomic-role parametre : domain (OR Fonction  
Contradiction Ressources) : range Parametre)
```

Cela signifie qu'un paramètre est un attribut caractérisant les instances de contradiction ou de fonction et que cet attribut est une instance particulière de la classe Parametre.

Le second exemple définit que considere est un attribut de la classe Ressources, et que cet attribut est un BOOLEAN, de valeur TRUE ou FALSE.

```
(define-atomic-role considere :feature true : domain Ressources  
: range Boolean)
```

La spécification `feature true` précise que cet attribut est unique, c'est à dire de cardinalité 0 ou 1.

Nous illustrons également la définition de deux concepts. Le premier est le concept Parametre, qui dérive du concept Modele et qui comporte deux attributs : `valeur_maxi` et `valeur_mini` qui sont des Float.

```
(define-concept Parametre  
  (AND Modele  
    (ALL valeur_maxi Float)  
    (ALL valeur_mini Float)))
```

Le second concept proposé est celui de Substance qui dérive directement du concept Ressources, Substance est une primitive de Ressources. Cela signifie qu'il n'existe pas d'attribut discriminant entre Substance et Ressources. Cette notion est définie par : `primitive true`.

```
(define-concept Substance Ressources:primitive true)
```

Il est normalement déconseillé de définir des primitives, nous avons ici créé les concepts de Ressources et de Modele afin de créer une distinction entre la modélisation des éléments du monde réel et les concepts relatifs à la modélisation du problème.

Une distinction entre la représentation centrée-objet dans Protege-2000 et la modélisation en logique de description est la nécessité, dans Prolog, de définir les types d'attributs (STRING, BOOLEAN, ...) en tant que concepts. Ces derniers sont déjà pré-définis dans Protege-2000.

La définition de l'ensemble des concepts permet de construire un graphe représentant la hiérarchisation des concepts et de valider la cohérence du modèle. La figure IV.2. représente la hiérarchisation des concepts issue de la modélisation en logique de description.

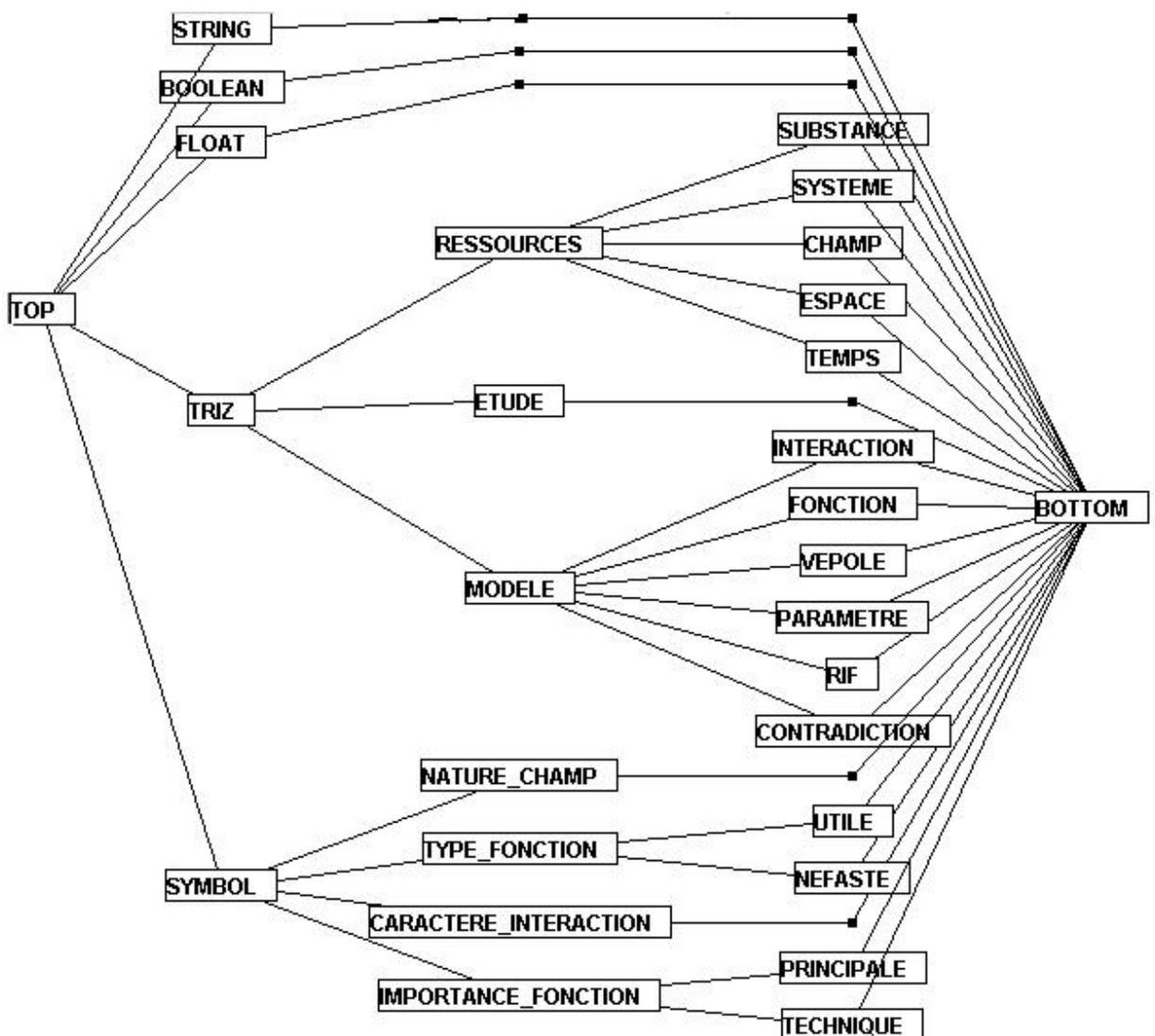


Figure IV.2. Hiérarchisation des concepts issue de la modélisation en logique de description, selon [Bultey, 2004]

III.2.2.3. Ecriture de règles d'inférences sur le modèle

L'objectif de cette partie est de définir des règles de connaissances expertes permettant d'exploiter les connaissances collectées, et d'inférer sur ces connaissances, afin de déterminer les caractéristiques du problème. Ceci est réalisé dans le but d'éviter à l'utilisateur d'avoir à connaître totalement la théorie.

Le système que nous utilisons pour définir les règles est un moteur de règles, Jess [Friedman-Hill, 2003] acronyme pour « Java Expert System Shell ». Inspiré du noyau de système expert CLIPS, Jess permet de définir des applets Java et des applications ayant la capacité de raisonner, en utilisant des connaissances fournies sous forme de règles déclaratives. [Mathieu, 1986] définit les systèmes experts comme des systèmes déductifs, i.e. réalisant des déductions, à partir d'un moteur d'inférence, sur une base de faits. Les faits sont alors les instances des concepts. La vision des systèmes de [Friedman-Hill, 2003] est un ensemble de règles, pouvant être appliquées à une collection de faits. Les connaissances, dans Jess, sont représentées de trois manières :

- les règles, qui représentent les connaissances heuristiques issues de l'expérience ;
- les fonctions, qui représentent les connaissance procédurales ;
- les frames permettent de définir les faits, c'est-à-dire les objets et les relations entre ces objets.

Les règles permettent de modéliser les connaissances expertes, et les fonctions de rendre un système expert opérationnel ; enfin, on a, à nouveau, le recours à la représentation centrée-objet pour la définition de la nature des connaissances à traiter. Notre modèle défini dans Protege-2000 pourra être directement utilisé grâce à JessTab, interface entre Protege-2000 et Jess, ; nous y reviendrons plus loin.

Les règles que nous avons définies ne sont que celles relatives à la formulation du problème selon le cadre modèle substance-champ (cf. Annexe-VII). D'autres règles expertes d'analyse du système, selon les lois d'évolution, devront être définies ultérieurement, leur écriture sort du cadre de cette thèse.

Quatre règles ont été définies afin de caractériser la nature d'un modèle substance-champ, selon qu'il est complet, incomplet, fonctionnel ou non-fonctionnel.

Nous présentons, ici, les règles relatives à la caractérisation d'un modèle substance-champ complet et d'un modèle substance-champ fonctionnel, dans le but d'illustrer la syntaxe de programmation et de clarifier, si besoin est, les notions.

Le modèle substance-champ complet se définit comme suit :

```
(defrule MAIN::Vepole_complet
  ?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (elements
  $?a ?s1&:(element-est-un Substance ?s1) ?s2&:(element-est-un
  Substance ?s2) $?b))
  (MAIN::object (is-a Vepole) (:NAME ?n) (elements $?w
  ?c&:(element-est-un Champ ?c) $?x))
  (not (MAIN::object (is-a Vepole) (:NAME ?n) (completude
  TRUE))))
=>
(slot-set ?vepole completude TRUE))
```

Cette règle stipule qu'un modèle substance-champ sera défini complet, i.e., l'attribut `completude` de `Vepole` sera mis à la valeur `TRUE`, s'il est composé de deux substances distinctes et d'un champ et qu'il n'est pas encore défini complet. Cette dernière condition est uniquement une condition d'optimisation du moteur d'inférence.

Le modèle substance-champ fonctionnel se doit d'être complet, mais il doit également être constitué d'une interaction réalisant une fonction utile, et qui soit à caractère satisfaisant. La définition de ces conditions se fait par la règle suivante :

```
(defrule MAIN::Vepole_fonctionnel
  ?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n)
  (fonctionnalite ?fonctionnel&:(neq ?fonctionnel TRUE))
  (interactions $?inter1 ?objet $?inter2) (completude TRUE))
  (MAIN::object (is-a Interaction) (OBJECT ?objet) (caractere
  ?caractere&:(eq (str-cat ?caractere) "Satisfaisante")) (fonction
  ?fonction))
  (MAIN::object (is-a Fonction) (OBJECT ?fonction) (type
  ?type&:(eq (str-cat ?type) "Utile"))))
=>
(slot-set ?vepole fonctionnalite TRUE))
```

La caractérisation du type de modèle substance-champ se fait automatiquement lors de l'instanciation du modèle. Elle permet d'orienter vers les règles de transformation du modèle adéquates.

III.2.2.4. Création d'une interface

L'ontologie est définie dans Protege-2000 et les règles heuristiques dans Jess ; il est donc nécessaire de créer une passerelle entre les deux. JessTab [Eriksson, 2003a] est une telle passerelle. JessTab autorise un mappage des bases de connaissances définies dans Protege-2000 en faits dans Jess ainsi que la manipulation de ces bases de connaissances [Eriksson, 2003b]. Ainsi, les règles de Jess peuvent inférer sur des instances de Protege-2000.

Le fonctionnement de l'interface d'aide à la formulation des problèmes, selon les cadres de la TRIZ, repose sur un questionnement qui oriente l'instanciation progressive des modèles. La progression du questionnaire dépend du type de problème à résoudre. Nous basant sur le critère d'idéalité, défini par l'équation :

$$I = \frac{\sum \text{fonctions utiles}}{\sum \text{fonctions néfastes} + \sum \text{fonctions coûts}}$$

nous avons déterminé quatre types de problèmes :

- la création d'un nouveau système
- l'ajout d'une nouvelle fonctionnalité à un système existant
- l'amélioration d'une fonctionnalité existante
- la suppression d'un effet néfaste dans un système existant.

L'ordre des éléments à collecter est orienté par la nature du problème à résoudre, comme illustré à la figure IV.3. pour l'amélioration d'une fonction utile. En effet, l'amélioration d'une fonction utile suppose d'avoir connaissance de ladite fonction, donc du produit sur lequel elle agit. A partir de ces points, peuvent être collectés le système réalisant la fonction ainsi que le paramètre du produit modifié par la réalisation de la fonction...

En revanche, la création d'un nouveau système demande d'identifier, en premier lieu, le produit sur lequel le système agira ainsi que le paramètre du produit à modifier, afin de définir correctement la fonction et la nature de l'outil qui agira sur le produit.

L'ensemble des questions est défini dans Protege-2000 (cf. Annexe-VIII). Elles sont caractérisées par la nature des réponses qui y sont apportées.

- Le type de réponse peut être simple, multiple ou aucune ; elle peut également être sans contrainte ou choisie dans une liste.

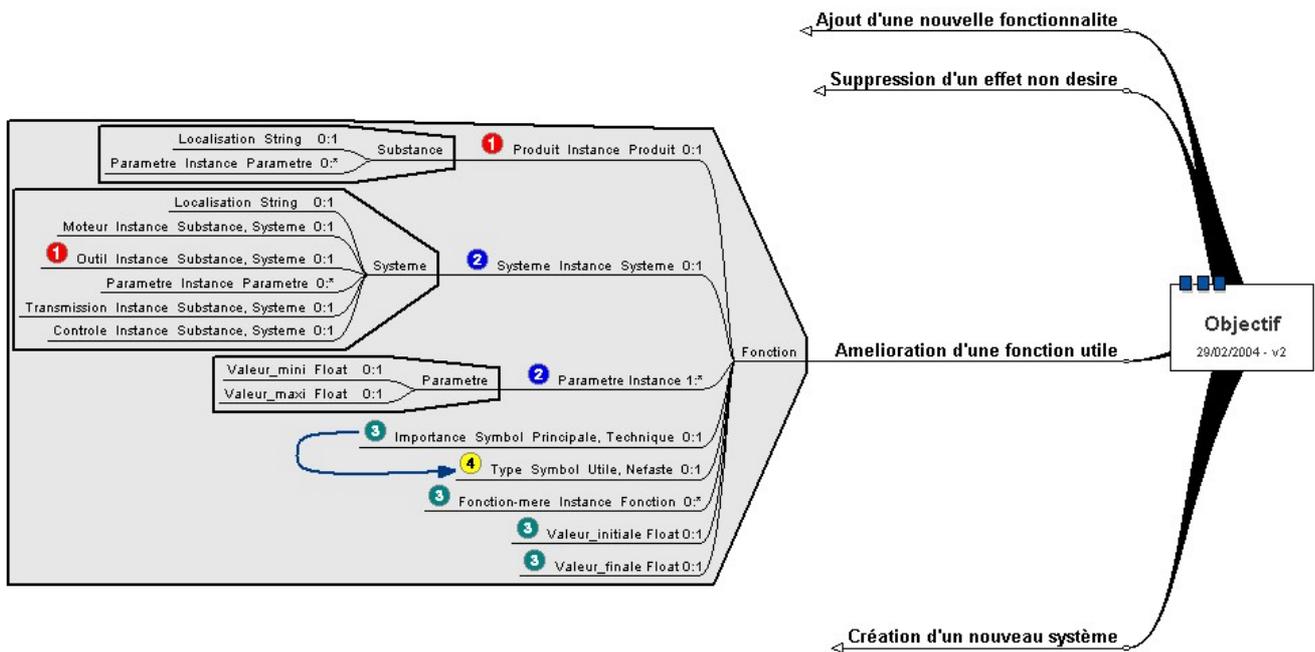


Figure IV.3. Stratégie d'instanciation des concepts

- La réponse peut être une instance particulière d'un concept ; par exemple, à la question : « Quelle est la fonction dont vous souhaitez améliorer la réalisation ? », la réponse sera une instance du concept Fonction, soit à choisir parmi l'ensemble des fonctions existantes, soit une nouvelle fonction à définir. Mais la réponse peut également être la spécification d'un paramètre d'une instance existante ; par exemple, la question : « Lors de la réalisation de " ?fonction ", un element de " ?nom " est en contact direct, agit directement sur " ?produit ". Quel est l'element en question ? » spécifie le paramètre outil de l'instance ?nom de systeme.
- La question requière une action spécifique du système qui aura, soit à modifier une instance existante, soit à créer une nouvelle instance ou ne rien faire.

L'enchaînement des questions se fait par l'existence d'instances du concept Autorisation. Ce concept autorise, pour une question donnée, et en fonction de la valeur de la réponse, la ou les questions que le système peut poser à l'utilisateur.

Les questions font souvent référence à des éléments précédemment collectés. Le moteur analyse alors les variables à collecter et ré-écrit le texte des questions en spécifiant la valeur de ces variables (l'ensemble des fonctions définies est donné en Annexe-IX).

L'interface est une fenêtre Java qui présente le texte spécifié de la question et dont les éléments dépendent de la nature de la question. Ainsi, pour une question simple, sera proposé un champ de texte `JTextField` pour la réponse, alors que pour les questions impliquant un choix dans une liste, est créé un élément `JList`. La figure IV.4. illustre l'interface avec une question à choix multiple.



Figure IV.4. Exemple d'interface

Le chapitre V. illustrera la mise en application du logiciel, par la présentation d'expérimentations menées pour valider l'utilité du modèle construit, dans l'aide à la formulation de problèmes selon les cadres de la TRIZ.

III.2.3. Conclusions sur le modèle

III.2.3.1. Proposition d'une sémantique claire et non ambiguë

L'étude a permis de clarifier les concepts qui participent à une bonne formulation des problèmes selon les cadres de la TRIZ. Cet ensemble de concepts, est maintenant structuré et lève les ambiguïtés de vocabulaire, qui existaient jusqu'alors. L'ensemble des concepts hiérarchisés est le suivant :

- Modèle
 - Paramètre
 - Fonction
 - Vépole
 - Interaction
 - Contradiction
 - Résultat Idéal Final
- Etude
- Ressources
 - Substance
 - Champ
 - Système
 - Espace
 - Temps

Il serait possible, sur la base de cet ensemble de concepts, de ré-écrire l'ensemble des textes de référence de la TRIZ, notamment ceux des standards. En effet ces derniers regorgent de descriptions non explicites, comme le recours aux notions de « Vépole » ou de « Système Vépole » pour désigner le modèle construit. La notion de complétude est précisée par la définition de la notion de fonctionnalité pour un modèle Vépole donné.

III.2.3.2. Eclaircissement du lien entre les modèles

Les concepts définis permettent d'identifier le lien entre les différents modèles de formulation de problèmes de la TRIZ, comme l'illustre la figure IV.1.

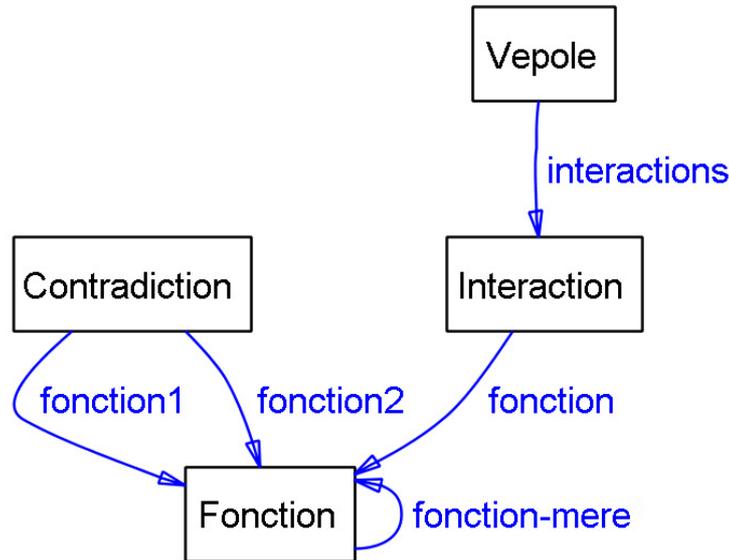


Figure IV.1. Liens entre les modèles, d'après la représentation centrée-objet

La figure IV.1. explicite le passage du modèle de la contradiction au modèle substance-champ. La contradiction résulte de l'opposition entre deux requis fonctionnels, réalisés par des interactions, lesquelles sont représentées dans le modèle substance-champ. Ces liens sont un apport important du modèle, permettant, en plus, de visualiser la possibilité de modéliser un problème, selon les cadres de la contradiction ou du modèle substance-champ, en partant de la représentation fonctionnelle.

III.2.3.3. Difficultés rencontrées

La construction du modèle s'est heurtée aux difficultés suivantes :

- Absence de réelle hiérarchisation des concepts identifiés, en effet, seules deux catégories de concepts sont définies : les concepts de formulation du problème (contradiction, modèle substance-champ, ...), qui regroupent les éléments des modèles, et les concepts représentant des éléments réels (substance, champ, système). Ces derniers sont les *Ressources*, ils incluent les descriptions spatiales et temporelles.
- Difficulté de définition du niveau de granularité du modèle : faut-il définir la fonction utile ou la fonction néfaste comme des concepts à part entière ? En considérant qu'un concept ne peut être défini que s'il est immuable, les rôles pris par les éléments ont été définis comme attributs de concepts. C'est le cas de l'utilité d'une fonction, car celle-ci peut varier d'un problème à un autre pour une même instance. Il en est de même pour les notions de moteur, outil, ... d'un système.

Par ailleurs, la tradition de transfert oral de la théorie est encore fortement ancrée dans les approches de la TRIZ qu'en ont les experts. Cela se traduit par une difficile acceptation du besoin de formalisation. Les experts ressentent la formalisation comme une fixation des concepts, or des concepts figés sont perçus comme limitatifs à l'approche de la conception inventive. Pour les experts, il n'est pas gênant que les concepts soient interprétés par l'utilisateur et que deux interprétations diffèrent, tant que cela permet d'atteindre le résultat. Notre approche pour valider la construction du modèle n'a donc pas été de le faire « valider », si la validation correspond au fait que le modèle construit soit unanimement accepté. Mais on a fait accepter le modèle en demandant si la construction proposée est « compatible » à la définition des concepts de la TRIZ.

IV. EXPLOITATION ET VALIDATION DE L'OUTIL

Ce chapitre fait le point sur les démarches de validation effectuées et sur celles restant à faire. Les expérimentations menées et présentées ici permettent en outre de montrer les apports et les limites de l'outil, ouvrant sur les perspectives d'amélioration et sur des thèmes de recherches à poursuivre.

IV.1. Méthode de validation

IV.1.1. Objectifs de la validation

L'outil proposé a pour objectif d'accompagner un utilisateur non expert de la TRIZ dans la formulation du problème à résoudre. C'est à dire, qu'il doit permettre à cet utilisateur, à partir d'une situation problématique, de dégager un problème clé à résoudre pour faire évoluer la situation. Ce problème clé sera formulé selon les cadres de la TRIZ pour les raisons déjà évoquées, d'intérêt dialectique de la contradiction et de possibilité de recours aux outils de la TRIZ pour synthétiser rapidement une voie générique d'évolution de la situation, comme le représente la figure IV.1..

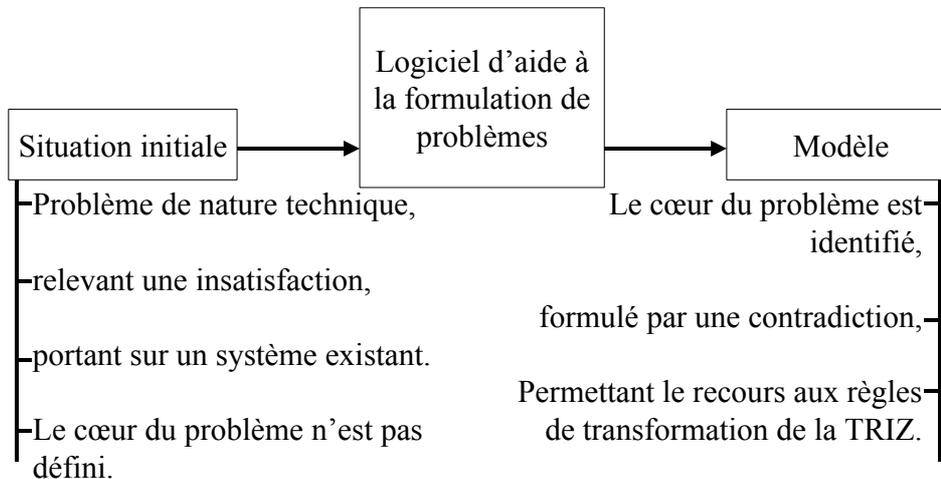


Figure IV.1. Entrants et sortants du logiciel d'aide à la formulation de problèmes

Les entrants et sortants décrits ne précisent pas que le logiciel, outre le fait d'accompagner la démarche de formulation du problème, doit également intégrer une notion pédagogique. L'utilisation répétée du logiciel doit tendre vers la formation de l'utilisateur. Cet aspect peut être rendu possible par la capitalisation des cas formulés et la possibilité de retravailler le parcours de formulation suivi.

Les objectifs cités permettent de définir une cible d'utilisateurs potentiels du logiciel, cette cible est caractérisée, d'une part, par le niveau des connaissances de la TRIZ, et, d'autre part, par le niveau des connaissances vis-à-vis du domaine d'apparition du problème. Ces deux axes sont représentés sur la figure IV.2..

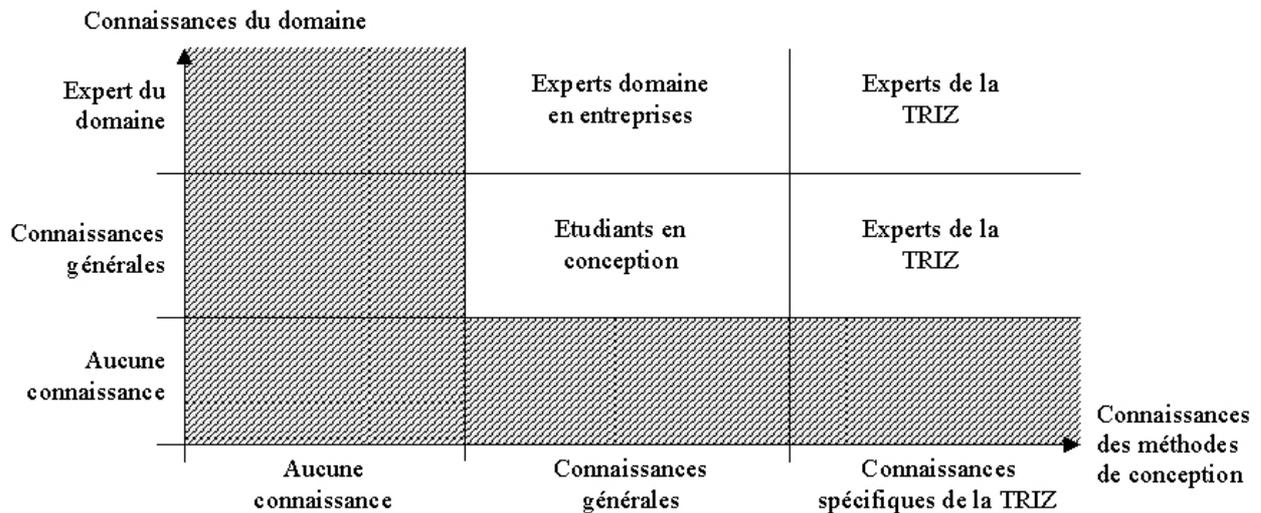


Figure IV.2. Cible des utilisateurs du logiciel

Les deux axes sont segmentés en trois niveaux. Pour l'axe connaissances du domaine du problème :

- Aucune connaissance spécifique ;
- Connaissances techniques générales ;
- Expert du domaine ;

pour l'axe connaissance des méthodes de conception :

- Aucune connaissance spécifique ;
- Connaissances de méthodes de conception, mais pas de connaissance particulière de la TRIZ ;
- Connaissances spécifiques de la TRIZ ;

Notre objet d'étude, la conception des systèmes techniques, implique que le public concerné ait un minimum de connaissances à la fois techniques et de connaissances sur les méthodes de conception. Les lignes et colonnes de cible n'ayant aucune connaissance techniques ou de méthode de conception n'ont, pour nous, aucun intérêt. Par ailleurs, il semble évident que les experts de la TRIZ n'ont pas besoin d'un tel outil pour formuler les contradictions, mais ils

pourraient y trouver un intérêt en vue d'une capitalisation des cas et, éventuellement, pour la possibilité de générer des comptes-rendus des études.

La cible visée est celle d'utilisateurs ayant des connaissances générales des méthodes de conception, sans expertise particulière de la TRIZ. Deux types de public ont été identifiés, le premier est celui d'étudiants formés à la conception ou d'un public ayant suivi une formation de base à la TRIZ. Ce premier public a une connaissance des notions de base de la TRIZ, mais est confronté à des difficultés dans sa mise en œuvre, notamment sur la phase de collecte des connaissances pour formuler les problèmes. Le second public est celui des experts d'un domaine, présents en entreprise, ayant à résoudre des problèmes de conception, mais n'ayant aucune connaissance spécifique de la TRIZ.

IV.1.2. Expérimentations de validation

Pour des raisons de facilité, notre procédure de validation a été, en premier lieu, de tester le logiciel sur la cible des étudiants. Notre démarche s'est déroulée en deux étapes sur deux groupes d'étudiants ayant eu une sensibilisation à la TRIZ :

1. Un groupe de quatre étudiants du DESS Production Automatisée, option conception intégrée, ayant suivi une initiation de 24 heures (cours et Travaux Dirigés) à la TRIZ, auquel est soumis un problème de type « cas d'école ». L'objectif de cette étape est de juger si l'utilisation du logiciel permet d'atteindre une formulation correcte du problème, c'est à dire de formuler une contradiction portant sur le paramètre clé du problème.
2. Un groupe de quatre étudiants de la maîtrise ATIBA (« Arts et Techniques Industrielles du Bois et de l'Ameublement »), option ameublement, ayant suivi une initiation de 24 heures (cours et Travaux Dirigés) heures à la TRIZ. Ces étudiants travaillent sur un projet qu'ils ont choisi, et ont utilisé le logiciel pour dégager le problème clé relatif à leur projet. L'objectif est ici de comparer l'apport du logiciel pour formuler la contradiction à une démarche en autonomie, suite à une brève introduction.

Nous allons illustrer, pour chaque étude, le cheminement de l'un des étudiants afin d'illustrer la démarche d'utilisation du logiciel et de présenter les conclusions que nous pouvons retirer de ces deux expérimentations.

IV.2. Illustration des expérimentations

IV.2.1. Validation de la convergence de la formulation

Le problème suivant est soumis à quatre étudiants :

« Lorsque l'on souhaite souder deux pièces ensemble, l'usage d'un fer impose de tenir celui-ci le plus proche de la pointe afin d'accroître la précision. Ce faisant, l'utilisateur s'approche de la chaleur et se brûle. Comment résoudre ce problème ? »

Les caractéristiques du problème qui nous paraissent intéressantes sont :

- C'est un problème de re-conception d'un système existant, il permet donc une bonne formulation de la contradiction. Nous évitons en effet les problèmes de détection, appartenant à une classe spécifique des standards de la TRIZ, pour lesquels la formulation de la contradiction est plus compliquée.
- Peu d'indications sur la situation initiale sont données, afin d'amener les étudiants à se poser les questions importantes, auxquelles les réponses seront apportées progressivement, par la définition des contraintes.
- Les étudiants stoppent l'utilisation du logiciel lorsqu'ils aboutissent à une formulation qui leur paraît refléter correctement la situation problématique.

L'objectif de cette étude est d'arriver à la formulation d'une contradiction physique qui soit une bonne représentation du problème pour l'étudiant. Nous connaissons à l'avance le paramètre clé du problème, qui est la longueur du manche.

Ci-dessous, nous présentons l'instanciation réalisée par un étudiant, qui a utilisé le logiciel durant trente minutes pour aboutir à une formulation le satisfaisant.

Donnez un nom au projet.

Problème du fer

Quel est l'objectif de l'étude ?

Amélioration d'un système existant.

Quel est le nom du système à améliorer ?

Le fer

Quelle est la fonction principale réalisée par " Le fer " ?

Souder

De quelle manière souhaitez-vous améliorer le système " Le fer " ?

Suppression d'un effet néfaste.

Quel est l'effet gênant, la fonction gênante, à supprimer ?
(formulation conseillée: un verbe + un COD)

Brûler l'utilisateur

La réalisation d'une fonction s'effectue dans le but de modifier une caractéristique d'un objet. Quel est l'objet en question, pour la fonction " Brûler l'utilisateur " ?

La main de l'utilisateur

" La main " est-il : (Substance, Système)

Substance

" Brûler la main " est la modification d'un paramètre en particulier de " La main de l'utilisateur ". Quel est ce paramètre ?

Sa chaleur

Quelle est la valeur de " Sa chaleur " avant l'exécution de " Brûler l'utilisateur " ?

faible

Quelle est la valeur de " Sa chaleur " après l'exécution de " Brûler l'utilisateur " ?

haute

Quelle est la cause de " Sa chaleur " " haute " ?

cela est la CONSÉQUENCE de la réalisation d'une fonction utile

Quelle est la fonction utile qui n'est pas assurée si " La température " n'est pas " haute " ?

La précision

Il faut donc résoudre la contradiction suivante: " Sa chaleur " doit être " haute " afin d'assurer la réalisation de " La précision ", mais " Sa chaleur " ne doit pas être " haute " afin d'éviter l'apparition de " Brûler l'utilisateur ".

Nous allons analyser le système responsable de l'apparition de " Brûler l'utilisateur ". Donnez un nom à ce système.

Le système de chauffage du fer

Lors de la réalisation de " Brûler l'utilisateur ", un élément de " Le système de chauffage du fer " est en contact direct, agit directement sur " La main ". Quel est l'élément en question ?

Le manche

Quel paramètre de " Le manche " peut-on modifier afin d'empêcher l'apparition " Brûler l'utilisateur " ?

La longueur

Quelle valeur " La longueur " de " Le manche " permet d'empêcher l'apparition de " Brûler l'utilisateur " ?

Long

Si " La longueur " de " Le manche " est a " Long ", le problème est-il résolu ?

Non

Quelle est la valeur actuelle pour " La longueur " de " Le manche " ?

Normale

Si nous ne pouvons le changer, c'est que le fait d'avoir " La longueur " de " Le manche " à " Normale " permet de réaliser une action utile. Quelle est cette fonction ?

La précision

La contradiction suivante est alors à résoudre: " La longueur " de " Le manche " doit être " Normale " pour assurer la réalisation de " La précision ", mais " La longueur " de " Le manche " doit être " Long " pour " Brûler l'utilisateur ".

Ces expérimentations ont mis en évidence des formulations de problème correctes dans trois des quatre cas réalisés. Le paramètre identifié a été à deux reprises celui de la longueur du manche, et à une reprise le pouvoir isolant du manche. Or, la solution proposée, par le brevet 916166 [Gasnov, 1995], porte sur l'utilisation d'un long manche permettant de séparer la pointe de la main par de l'air, qui est un très bon isolant. L'étudiant qui n'a pas abouti à une formulation satisfaisante présentait des lacunes en français, et n'arrivait que difficilement à comprendre les énoncés des questions.

L'aspect satisfaisant est que les étudiants sont parvenus à formuler correctement des contradictions en un temps relativement court (quarante minutes en moyenne). L'ensemble des expérimentations est fourni en Annexe-X.

IV.2.2. Validation de l'apport du logiciel à la formulation de contradiction

La deuxième partie de la validation a été menée sur des étudiants ayant choisis un problème à résoudre avec la TRIZ. Ils ont eu une initiation à la théorie puis ont travaillé pendant trois heures de travaux dirigés sur la formulation du problème. Ils sont alors venus utiliser le logiciel, afin d'envisager l'apport de ce dernier, soit pour valider leur formulation, soit pour ouvrir une nouvelle direction de recherche.

Nous présentons ici le cas d'un étudiant travaillant sur le système de serrage des pièces en bois massif sur pour usinage sur machine à commande numérique. Le problème, tel que l'étudiant le décrit, est : « Le positionnement est réalisé par des ventouses qui sont trop grosses pour les pièces en bois de petites dimensions. Il existe différentes valeurs de largeur et d'épaisseur des pièces à usiner, le démontage et le réglage des pièces se font manuellement. »

L'étude sans le logiciel a permis d'identifier six contradictions techniques. Voici le résultat de l'utilisation du logiciel :

Donnez un nom au projet.

Systeme de serrage des éléments en bois massif pour commande numérique

Quel est l'objectif de l'étude ?

Amélioration d'un système existant.

Quel est le nom du système à améliorer ?

système de maintien des pièces

Quelle est la fonction principale réalisée par " système de maintien des pièces " ?

maintenir les pièces

De quelle manière souhaitez-vous améliorer le système " système de maintien en position des pièces " ?

Amélioration d'une fonction utile.

Quelle est la fonction dont vous souhaitez améliorer la réalisation ?

réglage des vérins

Cette fonction est-elle décomposable ?

Non

Parmi les fonctions suivantes, quelle est celle dont la réalisation pose le plus de problèmes ?

réglage des vérins

La réalisation d'une fonction s'effectue dans le but de modifier une caractéristique d'un objet. Quel est l'objet en question, pour la fonction " réglage des vérins " ?

les vérins

"les vérins " est-il : (Substance, Système)

Substance

"réglage des vérins " est la modification d'un paramètre en particulier de " les vérins ". Quel est ce paramètre ?

la localisation

Quelle est la valeur de " la localisation " avant l'exécution de " réglage des vérins " ?

présent

Quelle est la valeur de " la localisation " après l'exécution de " réglage des vérins " ?

non présent

Il faut améliorer la réalisation de " réglage des vérins ". Quelle serait la valeur idéale du paramètre " la localisation " de " les vérins " après la réalisation de " réglage des vérins " ?

au contact de la pièce

Quel paramètre de " les vérins " faut-il modifier afin d'avoir " la localisation " à la valeur " au contact de la pièce " ?

la course des vérins

Quelle est la valeur " la course des vérins " de " les vérins " qui permet de mettre " la localisation " a " au contact de la pièce " ?

grande

Si " la course des vérins " de " les vérins " vaut "grande". Cela crée-t-il un effet gênant ?

Oui

Si on considère que l'effet gênant est la réalisation d'une fonction non souhaitée. Quelle est cette fonction ?

positionner le vérin

Il faut donc résoudre la contradiction suivante: " la course des vérins " de " les vérins " doit être "grande" afin d'avoir " la localisation " de " les vérins " a "grande". Mais " la course des vérins " de " les vérins " ne doit pas être "grande" pour éviter l'apparition de " positionner le vérin ".

L'utilisation du logiciel a permis de formuler une contradiction physique représentative d'un problème majeur. La voie proposée n'avait pas été préalablement étudiée.

Les quatre expérimentations menées ont abouti à des formulations jugées très intéressantes par les étudiants et permettant à certains de dégager des concepts de solution, dès la formulation de la contradiction. La durée moyenne d'utilisation du logiciel a été de quarante minutes, ce qui est intéressant par rapport aux trois heures passées par les étudiants pour identifier des contradictions techniques. Les expérimentations sont à consulter en Annexe-XI.

Nous sommes donc satisfaits de l'efficacité des heuristiques d'instanciation du modèle.

IV.3. Limites

Nous allons aborder dans cette partie les éléments limitatifs du logiciel, nous permettant de dégager des perspectives d'améliorations pour nos recherches futures. Les limites analysées portent sur les points suivants :

- la validation de l'efficacité du modèle sur l'ensemble de la cible visée ;
- l'interface ;
- l'intégration de l'ensemble des éléments de la TRIZ.

IV.3.1. Limites de validation

Nous avons mené à bien les validations concernant la cible ayant à la fois des connaissances générales en ingénierie et en méthodes de conception. Comme précisé auparavant, il serait intéressant de mettre le logiciel à disposition d'experts ayant à résoudre des problèmes dans leur domaine d'excellence. En effet, leur statut d'expert d'un domaine signifie qu'ils sont en mesure de construire de manière efficace des représentations variées d'une même situation. Si, par le biais du logiciel, nous pouvons leur apporter une nouvelle approche des problématiques à résoudre dans leur domaine, et qu'ils jugent cette approche intéressante, cela constituera un apport indéniable.

Cette expérimentation pourra être effectuée lors du démarrage d'une étude par les experts de la TRIZ du laboratoire. Il paraît en effet intéressant d'insérer l'utilisation du logiciel au démarrage de l'étude, avant que les acteurs du projet ne soient influencés par les formulations des contradictions qu'ils seront amenés à formuler avec les experts.

Toutefois, avant de mettre en œuvre ces expérimentations, nous souhaitons au préalable améliorer l'interface du logiciel, afin de proposer un outil à l'aspect professionnel.

IV.3.2. Limites de l'interface

L'interface graphique du logiciel est à améliorer. Pour l'instant nous avons défini un prototype nous permettant de valider le contenu du logiciel. Les fenêtres définies jusqu'à présent se limitent à permettre l'utilisation des questions, afin d'instancier le modèle. Plusieurs améliorations sont à apporter. En premier lieu, il sera nécessaire de compiler correctement le logiciel afin de faciliter l'installation sur d'autres postes. Celle-ci passe aujourd'hui par les installations successives de Protege-2000, de Jess puis de JessTab, ce qui n'est ni convivial ni intuitif. Dans un second temps il sera nécessaire de rendre Protege-2000 transparent à l'utilisation du logiciel. En effet, le logiciel est, pour l'instant, utilisé à partir de l'onglet de commande JessTab sous Protege-2000.

L'interface, telle que nous l'envisageons à terme, dans le but d'accroître à la fois la convivialité d'utilisation et l'explicitation de la démarche devra comporter un certain nombre d'améliorations.

- La création d'un journal de suivi des questions posées permettra d'accéder directement à une question antérieure et d'en modifier la réponse. Pour l'instant les retours se font via un bouton « Question précédente ».
- L'implantation d'une représentation graphique du modèle de la contradiction et du modèle substance-champ doit permettre la compréhension du rôle des éléments de connaissance collectés, par rapport au modèle de problème.
- L'implantation d'une fenêtre de visualisation des éléments existants dans la base de connaissances (les systèmes définis au préalable, les fonctions qui y sont liées), facilitera la réutilisation des éléments définis dans les études antérieures. Cette partie est relative à l'utilisation du logiciel pour constituer une base de connaissances métier.

L'apport de ces améliorations va être entrepris, la programmation d'interfaces conviviales peut être réalisée par le recours aux composants du package « Swing » de Java. L'implantation des représentations graphiques des problèmes peuvent s'appuyer sur les représentations traditionnelles reprises sur la figure IV.3.

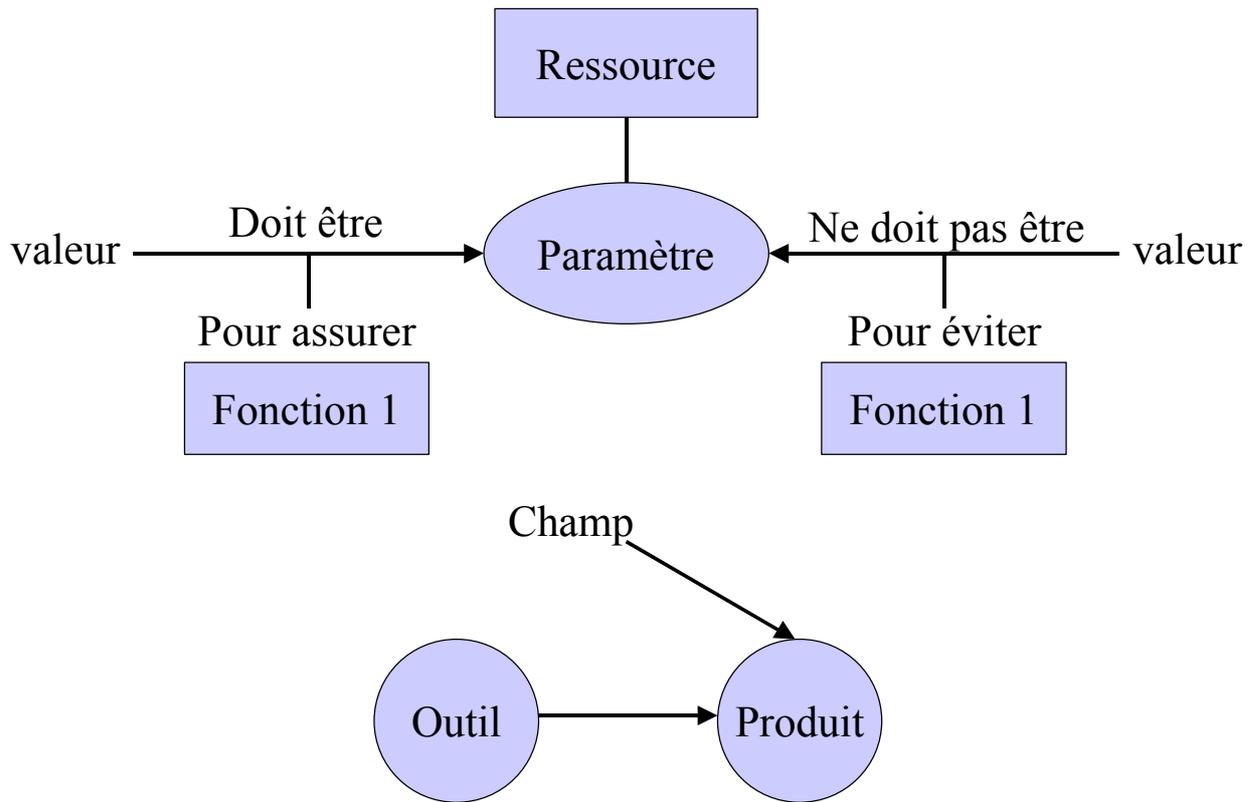


Figure IV.3. Représentations graphiques des modèles de problèmes

Enfin, il est nécessaire de rendre les corrections des réponses apportées antérieurement plus conviviales et robustes. En effet deux types de questions co-existent dans le programme, celles liées à l'interface d'instanciation du modèle, qui participent à la formulation du problème et sont issues de la stratégie de formulation. D'autre part des questions sont posées automatiquement, en tant que règles issues de Jess lorsqu'un élément est défini mais non complètement renseigné. Ce second type de question rend les retours en arrière non robustes, ce défaut est à corriger.

IV.3.3. Limites du modèle vis-à-vis de la TRIZ

Comme précisé initialement, l'objet d'étude de cette thèse est axé sur les démarches et modèles de formulation de problèmes de la TRIZ. L'apport de ce travail, notamment en termes de formalisation, doit être étendu à l'ensemble de la théorie. Des éléments sont entrepris, avec la ré-écriture des règles standards de transformation des modèles substance-champ, sur la partie résolution des problèmes. Deux manques importants restent à combler :

- La résolution des problèmes passe par l'identification d'une ressource solution. La démarche de résolution de la TRIZ est la caractérisation d'un élément de solution par la définition des paramètres de cette solution permettant de résoudre la contradiction. Reste alors à reconnaître, au travers de ces caractéristiques un élément du monde pouvant satisfaire ces caractéristiques. La théorie propose des règles de recherche de la ressource : tout d'abord au sein du système, puis dans les systèmes adjacents et, enfin, recourir à toutes les ressources existantes. Cette phase, qui consiste à identifier un élément du monde réel à partir d'une métaphore de la solution n'est pas du tout assistée. La difficulté est que les paramètres de caractérisation d'un élément solution ne sont pas connus a priori. Ainsi, pour automatiser cette étape il faudrait générer un moteur capable de décrire tout élément selon tout paramètre de description. Ce champ relève d'une contradiction clé non résolue en Intelligence Artificielle, à savoir que la modélisation est nécessaire pour raisonner sur la réalité, mais la modélisation n'est pas à entreprendre car elle ne représente qu'une partie de la réalité, et pas toujours celle nécessaire pour arriver à des conclusions.
- L'intégration des lois d'évolution est à faire. En effet, celles-ci sont une partie essentielle de la théorie dont la compréhension n'est pas triviale. Un travail d'identification et de définition des concepts relatifs aux lois d'évolution est nécessaire, afin d'étendre l'ontologie et d'écrire des règles d'inférence formalisées. Ce travail peut aboutir à une facilitation de la prise en compte du niveau d'évolution des systèmes et des tendances d'évolution à privilégier. Par exemple, la description des systèmes est construite selon la loi d'intégrité des parties. Cela signifie qu'il est possible de représenter un système selon ce cadre, mais la loi n'est pas exploitée, car l'analyse du rôle joué par chaque partie ne figure pas dans la stratégie développée. Sur la base de cette description des systèmes, un module d'analyse selon les lois d'évolution peut être envisagé.

IV.4. Apports et perspectives

Nous allons détailler ici les apports de nos travaux, en les positionnant par rapport aux problématiques énoncées dans le paragraphe II.5. Tout d'abord l'apport technique concernant la création d'un logiciel d'assistance à la formulation de problèmes, puis le positionnement scientifique de l'apport vis-à-vis de la formalisation de la TRIZ. Ces apports nous permettent, en outre, d'ouvrir les perspectives de développement de nos travaux.

IV.4.1. Apport technique : assistance à la formulation de problèmes

Nous rappelons la problématique industrielle, à laquelle nous souhaitons apporter une réponse :

Construire un outil de conception assistée par ordinateur, pour guider l'utilisateur dans la mise en œuvre des techniques de la TRIZ, en centrant nos travaux sur la construction d'un modèle de problème, selon les cadres de la TRIZ.

L'opérationnalisation de l'ontologie construite permet de guider les utilisateurs dans la formulation des problèmes, selon les cadres de la TRIZ. Certes, au vu des limites présentées précédemment, l'interface reste à améliorer, mais les expérimentations menées avec notre prototype ont montré un réel apport de l'utilisation du logiciel.

En effet, la mise en œuvre d'ARIZ, seule méthode actuellement formalisée de construction de la contradiction physique, est reconnue comme lourde et réservée aux seuls experts de la TRIZ. Le logiciel développé permet, à des non experts, de formuler une telle contradiction physique, qui plus est, dans un temps relativement court.

L'objectif de facilitation de la mise en œuvre des techniques de la TRIZ est atteint, pour les parties concernant la formulation des problèmes, selon les cadres de la contradiction physique et du modèle substance-champ.

IV.4.2. Apport scientifique : formalisation de la TRIZ

L'objectif scientifique auquel nous nous sommes proposés de répondre est le suivant :

Formaliser les éléments constitutifs de la TRIZ pour définir les concepts inhérents à la théorie et expliciter les liens entre ces concepts, en centrant nos travaux sur la construction d'un modèle de problème, selon les cadres de la TRIZ.

La démarche entreprise a permis de préciser l'ensemble des concepts, tels qu'ils sont définis dans la littérature sur la TRIZ, inhérents à la formulation des problèmes. Ces concepts sont cohérents avec les éléments de connaissance génériques à collecter pour la formulation de problèmes, selon les approches de l'Intelligence Artificielle. Nous avons, au travers de cette analyse un double but. D'une part, nous souhaitons asseoir la position de la TRIZ comme une théorie s'attachant à la résolution des problèmes pour aborder la conception. D'autre part, permettre une meilleure transmission des savoirs de la TRIZ. Ces deux objectifs passent par une réalisation commune, qui est la formalisation de cette théorie.

Notre travail d'identification, d'explicitation et de modélisation des concepts et des liens entre les concepts a permis d'aboutir, tout au moins sur les parties de la théorie traitant de la formulation des problèmes, à un ensemble homogène et cohérent. La validation de l'homogénéité et de la cohérence entre les concepts a pu se faire en s'appuyant sur une méthodologie que nous avons mis au point sur la base des outils d'Intelligence Artificielle existants.

Nous avons du établir notre propre démarche et recourir à plusieurs outils pour répondre aux spécificités de notre étude qui sont :

- peu de textes traitant de cas résolus, en tous cas pas assez pour couvrir l'ensemble de la problématique de formulation des problèmes
- les textes théoriques existants traitent de concepts génériques et s'appuient sur des modèles déjà existants mais peu formalisés

Nous disposons maintenant d'une base de description des concepts de la TRIZ permettant de présenter clairement les concepts manipulés, dans les phases de formulation de problèmes, et de définir une terminologie non ambiguë.

IV.4.3. Perspectives de développement

Comme précisé au travers des limites, deux voies de poursuite des travaux présentés dans cette thèse se dégagent. Dans un premier temps, l'élargissement des travaux à l'ensemble de la théorie, puis, dans un second temps, à la résolution des problèmes non techniques en s'appuyant sur les études issues d'OTSM-TRIZ.

IV.4.3.1. Intégration de l'ensemble des techniques au logiciel d'assistance à la conception

Nous avons déjà abordé la problématique d'intégrations de l'ensemble de la théorie à nos travaux dans les limites des travaux actuels. Nous rappellerons simplement ici les deux voies principales d'élargissement des travaux :

- la prise en compte des lois d'évolutions ;
- l'intégration des techniques de recherche de la ressource à utiliser pour synthétiser la solution spécifiée.

IV.4.3.2. Création d'un logiciel d'assistance à la formulation de problèmes

IV.4.3.2.1. Présentation d'OTSM-TRIZ

La TRIZ est basée sur les lois d'évolution des systèmes techniques, de fait elle s'intéresse à la résolution de problèmes techniques. Toutefois, les méthodes et cadres de formulation synthétisés dans la théorie sont applicables pour la résolution de problèmes non techniques. De nombreux experts de la TRIZ ont mis en œuvre ces méthodes dans des domaines aussi variés que la pédagogie, l'art ou même la politique.

Les travaux du projet Jonathan Livingston, menés par Khomenko ([Khomenko, 2001; Khomenko and Sokol, 2000]) abondent dans le sens d'une théorie de résolution des problèmes, quel que soit le domaine d'apparition de ces problèmes. Ainsi le cadre de formulation des problèmes est la contradiction, mais les différents niveaux de la contradiction ont été retravaillés.

La solution à un problème est décrite, dans OTSM-TRIZ, par le fait que « *des paramètres spécifiques, d'éléments spécifiques, ont des valeurs spécifiques pour une situation spécifique* » [Khomenko, 2000b]. Issue du besoin d'élargir les travaux de la TRIZ à la résolution générale des problèmes, l'approche OTSM-TRIZ repose sur un ensemble d'axiomes [Kucharavy, 2001a] donnés ci-dessous.

- L'axiome de la description définit que toute pensée s'appuie sur des modèles des éléments du monde réel et non sur les éléments eux-mêmes. Ainsi n'est prise en compte qu'une vue partielle des éléments et non la totalité, dans la description des problèmes.
- Les axiomes de la pensée sont au nombre de trois. Le premier précise que le cœur du problème provient d'une contradiction entre une loi objective et un souhait d'évolution lié à une situation spécifique. Le second indique que la résolution du problème passe par l'identification d'une impossibilité qui représente la difficulté du problème. Enfin, l'axiome de réflexion impose de changer de point de vue pour parvenir à contourner l'impossibilité relevée.
- Les axiomes de la vue du monde définissent le monde comme un tout unifié et régi par un ensemble de lois, appelées les « lois objectives générales ». Ce tout est composé d'éléments particuliers, eux-mêmes régis par des lois objectives particulières. Ces lois générales et particulières ne sont pas indépendantes, car elles sont reliées par les ressources composant le monde. Ces ressources doivent être considérées comme des processus, du fait qu'elles évoluent dans le temps.

Quatre principales techniques de mise en œuvre de ces axiomes sont en cours de conceptualisation, correspondant à l'identification de la cause du problème, à la mise en œuvre de solutions typiques de résolution, au recours aux contradictions pour les problèmes non standards, puis à la synthèse d'une solution issue d'une décomposition d'un problème complexe en un flux de problèmes.

IV.4.3.2.2. Mode d'intégration d'OTSM-TRIZ

L'approche OTSM-TRIZ est issue des travaux sur la TRIZ, et notamment des modes de représentation des problèmes. Ainsi la formalisation que nous proposons est une base à l'explicitation des concepts de OTSM-TRIZ. Il nous faut distinguer dans notre modèle les éléments qui sont génériques de ceux qui sont spécifiques aux problèmes techniques.

Nous pouvons, a priori, supposer que certains concepts n'auront plus de sens, comme celui de modèle substance-champ ; tandis que d'autres seront peut être à re-définir de manière plus large. Ainsi le concept de fonction, ayant une forte connotation technique, pourrait être re-défini comme la « cause » ou l' « origine » d'une valeur contradictoire.

Le modèle substance-champ repose sur des interactions entre substances dont la nature est un champ mécanique, thermique, chimique, électrique ou magnétique. Dans un problème

organisationnel ou politique, par exemple, les interactions sont de nature humaine et l'on ne peut modéliser des être humains, aux réactions complexes et aux variables multidimensionnelles (dimensions psychologiques, émotionnelles, etc.), de la même manière que l'on modélise des substances. De même, dans les problèmes relatifs aux systèmes d'informations, les interactions sont, justement, des informations, les dimensions sémantique et sémiotique sont alors à prendre en considération. Ces exemples illustrent que, sorti du domaine technique, de nouvelles dimensions sont à considérer, rendant le modèle substance-champ non applicable. La « philosophie » du modèle, basé sur la caractérisation des interactions entre les éléments du modèle, et sur la caractérisation de ces éléments, semble toutefois pertinente, et reste à développer de manière spécifique dans chaque domaine.

Les notions de « fonction » ou de « modèle substance-champ » seront à remplacer par des concepts plus abstraits, subsumant les notions précédentes. La nature de ces concepts abstraits pourra être établie par la généralisation d'un ensemble de concepts spécifiques issus de divers domaines.

IV.4.3.3. Développement d'une démarche de raisonnement à partir de cas

Le raisonnement à partir de cas est largement répandu en conception [Kolodner, 1993]. Ce type de raisonnement peut être vu aussi bien comme une méthode pour résoudre des problèmes que comme une méthode pour capturer de nouvelles expériences [Perner, 2002]. La figure IV.4. illustre ces deux cheminements. Lorsqu'un nouveau problème est à résoudre, sa similarité est mesurée avec les cas présents dans la base de cas. Si une similitude satisfaisante est identifiée, le cas cible, issu de la base, permet de générer le concept, ou métaphore, de la solution. Dans le cas contraire, il est nécessaire de résoudre le problème par un processus de re-formulation qui permet de construire un modèle du problème qui isole les caractéristiques clés du problème. Cette abstraction peut alors être, une nouvelle fois, confrontée à la base de cas, afin de générer une métaphore de la solution. Si aucune similarité n'est reconnue, le problème est original et doit être résolu de manière autonome, sans recours à la base de cas. Il permet alors de maintenir cette base et de l'enrichir d'un nouveau cas. Ce processus d'apprentissage comprendra la description du problème initial, du modèle de problème construit, et de la solution.

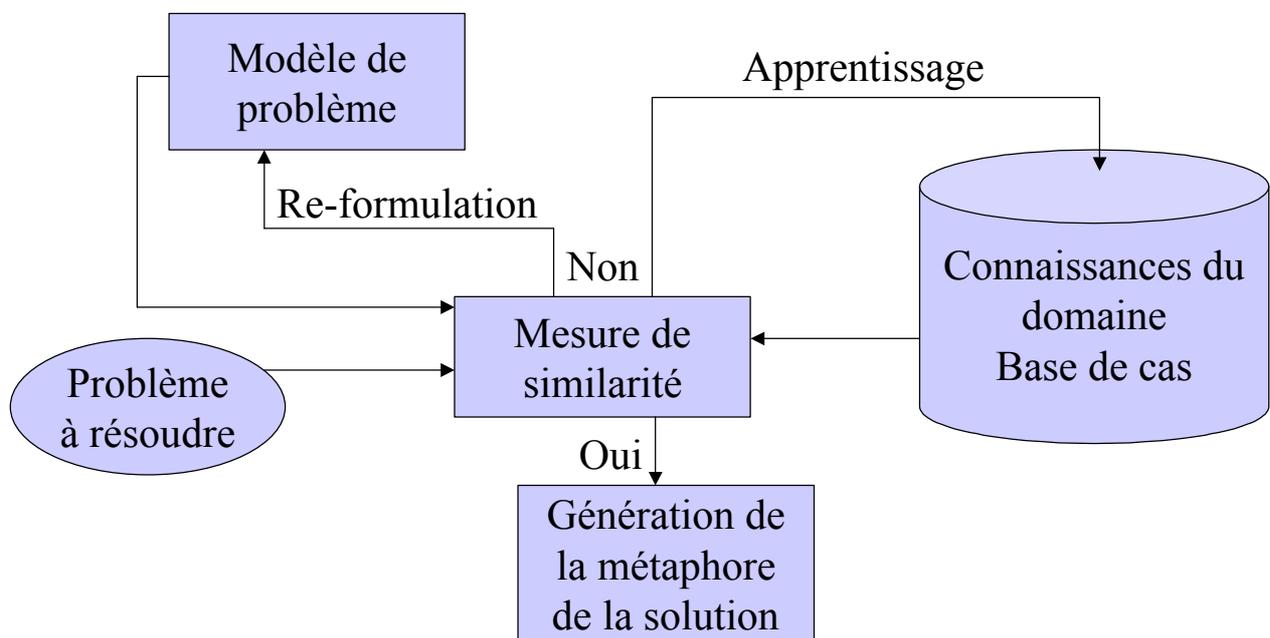


Figure IV.4. Le raisonnement à partir de cas

L'outil que nous avons conçu est intégré dans la constitution d'un outil de raisonnement à partir de cas. L'ontologie proposée fournit le modèle de problème et permet de mesurer la similarité entre les cas par l'identification de fonctions communes, d'un paramètre clé du problème commun ou d'un modèle substance-champ identique. Le modèle fournit, de plus, le

mode de capitalisation des connaissances dans la phase d'apprentissage. Enfin, le logiciel permet d'accompagner la re-formulation, quand celle-ci est nécessaire.

Afin de développer un outil de raisonnement à partir de cas, il reste à développer un mode de saisie de la situation initiale qui permette de reconnaître au plus tôt les similitudes. En outre, le moteur d'appariement, qui calcule les similitudes et propose une métaphore de solution appropriée est à établir.

L'identification des ressources utilisées pour la résolution des problèmes permet de construire progressivement une « ontologie TRIZ » du domaine. Cette modélisation des connaissances du domaine sera contextualisée par la caractérisation des paramètres des ressources qui participent à la résolution des contradictions. C'est pour quoi nous l'appelons une « ontologie TRIZ ».

IV.5. Conclusion

Le logiciel développé sur la base de l'ontologie est à l'état de prototype, comme nous l'avons montré précédemment. Toutefois ce prototype opérationnel nous a permis de valider le contenu et la démarche d'instanciation. Il offre, de plus, de réelles perspectives de développement. Nous avons atteint les objectifs que nous nous étions fixés, tant au niveau de la problématique industrielle que de la problématique scientifique.

Le développement d'une interface conviviale, conjuguée à une compilation propre, va permettre de distribuer à plus grande échelle le logiciel afin d'avoir un maximum de retours pour l'améliorer en continu.

Le développement du logiciel par l'intégration progressive des différentes techniques de la TRIZ nous permettra de développer un outil d'assistance pour tout le processus de génération de concepts en phase de conception. Cette diffusion pourra se faire par une utilisation systématique du logiciel dans les projets d'étudiants devant faire face à de la résolution de problèmes techniques, puis dans un second temps avec les entreprises partenaires du laboratoire.

Enfin, l'étude des techniques de OTSM-TRIZ et leur intégration logicielle auront pour double objectif de formaliser ou, le cas échéant, de développer cette théorie de résolution des problèmes non techniques, issue de la TRIZ.

CONCLUSION GENERALE

Une recherche à la frontière de deux domaines

Les travaux présentés dans cette thèse sont à la frontière de deux disciplines, la conception et l'Intelligence Artificielle. Le dénominateur commun, qui nous a permis de rapprocher ces disciplines, est la résolution des problèmes. Dans ce cadre, et pour alimenter nos réflexions sur la formalisation d'un processus de formulation de problèmes, en phase de conception, nos travaux se sont appuyés sur les démarches de modélisation des connaissances de l'Intelligence Artificielle, et sur les typologies de connaissances liées à la résolution de problèmes.

Le champ d'application de la démarche devait répondre à la fois :

- à une problématique industrielle : la forte croissance de la demande par rapport à la TRIZ, et les remarques sur les difficultés de formulation des problèmes, selon les cadres de la TRIZ, allaient en ce sens ;
- à une problématique scientifique : les nombreuses critiques formulées sur le manque de formalisation de la TRIZ et sur la non existence de méthodes explicites montraient les limites actuelles de cette théorie pour obtenir une validité scientifique.

Notre travail de thèse a pu s'appuyer sur des compétences centrées sur les problématiques de la conception inventive. L'appui d'experts de la TRIZ a notamment été un apport incontournable. Les compétences centrées sur la modélisation de connaissances, notamment à travers l'approche linguistique de modélisation d'un domaine, ont rendu ce travail possible.

En effet, la constitution d'ontologies, la construction du modèle des connaissances décrivant un domaine, est une expertise forte de la part des personnes travaillant en Intelligence Artificielle, qui développent des outils pour extraire ces connaissances et les expliciter.

Des résultats encourageants

L'analyse modulaire du processus de conception, présentée au chapitre 1, montre l'intérêt d'une approche globale des outils liés à la conception. Cet état de l'art permet notamment de montrer la potentielle complémentarité des diverses approches. Ainsi, le besoin d'homogénéiser le vocabulaire de la conception apparaît comme primordial. Une ontologie de la conception, qui permette de couvrir et mettre en relation l'ensemble des approches, est à construire.

Les résultats de cette recherche répondent aux attentes définies par les problématiques. La démarche issue de l'Intelligence Artificielle permet de formaliser de manière précise et sans ambiguïté les concepts inhérents à la formulation de problèmes, selon les cadres de la TRIZ. Le modèle ainsi constitué permet de ré-écrire et préciser les textes de référence. Bien sûr, l'objectif n'est pas de ré-écrire l'ensemble des textes parus, mais tout au moins les parties opératoires, relatives aux méthodes de formulation et de résolution des problèmes. En outre, elle fournit un cadre de formalisation clair pour les publications futures.

L'apport en terme de mise en œuvre, la formalisation d'une méthode de formulation permettent de faciliter à la fois l'utilisation de la TRIZ et sa transmission. Si l'aspect pédagogique est encore à renforcer, on sent le potentiel pour la constitution d'un apprentissage par l'expérimentation. L'utilisation de la TRIZ se trouve facilitée dans sa phase d'identification des problèmes à résoudre et de formulation de ceux-ci, de manière à pouvoir recourir aux outils de résolution de la TRIZ.

Toute la phase de génération de concepts de solution peut donc être assistée la prochaine étape étant de compléter cette assistance pour la phase de spécification du concept de solution.

La recherche pluridisciplinaire, source d'intégration

Les travaux de cette thèse montrent l'intérêt de développer des recherches pluridisciplinaires et la question peut se poser de l'élargissement de ces travaux aux autres domaines de compétences qui composent le génie industriel. Ne serait-il pas intéressant d'y intégrer les connaissances issues des sciences de l'éducation ? La constitution d'un logiciel pédagogique impose de tenir compte des modes de compréhension et d'apprentissage des individus. De fait, le développement de l'interface du logiciel ne pourra se faire sans recourir à ces connaissances.

De même, notre démarche de formulation des problèmes n'intègre que les données techniques du problème. Nous favorisons ainsi l'inventivité, mais, le succès d'une invention, qui

permet de transformer cette invention en innovation, impose d'intégrer des données sur les contraintes issues de l'entreprise :

- Dans quelle mesure l'entreprise peut-elle intégrer une invention ?
- Quelles ressources, financières, technologiques et humaines, l'entreprise peut-elle mobiliser pour développer l'invention ?
- Quelle réactivité l'entreprise a-t-elle ? Sur quel délai peut-elle développer l'invention ?

Les contraintes peuvent également émaner du marché :

- La cible est-elle prête à accueillir l'invention ?
- Comment former la cible d'utilisateurs à un nouveau mode d'utilisation ?
- Quelle peut-être le retour sur l'investissement ?

L'intégration de ces données impose d'étendre le cadre de formulation de problèmes, mais avant tout, de pouvoir identifier les caractéristiques de ces contraintes et leur mode d'intégration dans la définition du problème.

Nous voyons que la recherche que nous avons entreprise sur la formulation des problèmes, dans le cadre de la conception, s'inscrit dans une thématique très large qu'est la gestion de l'innovation. Nous avons conçu un module qui, dans l'état actuel, fonctionne indépendamment de son environnement ; il est nécessaire de l'inscrire dans le système « innovation » ; pour cela il faudra comprendre avec précision les liens entre l'activité de conception inventive et la gestion de l'innovation.

BIBLIOGRAPHIE

- [AFAV, 1994] AFAV (1994) *L'Analyse de la Valeur*, Ministère de la Coopération, Paris.
- [Akao, 1990] Akao, Y. (1990) *Quality Function Deployment: Integrating Customer Requirements into Product Design*, 0-915299-41-0, Cambridge, MA.
- [Altshuller, 1973] Altshuller, G. S. (1973) Step-by-Step. Dans *The Innovation Algorithm*, Technical Innovation Center, Worcester, Massachusetts, pp. 103-120.
- [Altshuller, 1975] Altshuller, G. S. (1975) *Processus de Résolution d'un Problème d'Innovation: Principaux étapes et mécanismes*, Traduit du russe par Julie Stien, LRPS, Strasbourg, 06/04/1975.
- [Altshuller, 1977] Altshuller, G. S. (1977) *Lois d'Evolution des Systèmes Techniques*, Traduit du russe par Julie Stien, LRPS, Strasbourg, 20/01/1977.
- [Altshuller, 1980] Altshuller, G. S. (1980) *Cinq Règles de Base, Des Ailes pour Icare*, Traduit du russe par Julie Stien, LRPS, Strasbourg.
- [Altshuller, 1986] Altshuller, G. S. (1986) *Histoire d'évolution d'ARIZ*, Traduit du russe par Julie Stien, LRPS, Strasbourg.
- [Altshuller, 1988] Altshuller, G. S. (1988) *Creativity as an Exact Science*, Gordon and Breach, 0275-5807, New York.
- [Altshuller, 1999] Altshuller, G. S. (1999) *The Innovation Algorithm: TRIZ, systematic innovation and technical creativity*, 0-9640740-4-4, Worcester, MA.
- [Altshuller and Shapiro, 1956] Altshuller, G. S. and Shapiro, R. B. (1956) Psychology of Inventive Creativity. *Issues of Psychology*, **6**, pp. 37-49.
- [Anderson et al., 2001] Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., R., P. P., Raths, J. and

-
- Wittrock, M. C. (2001) *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives*, Longman, Pearson Education, 0-32108-405-5, New York.
- [Antonietti et al., 2000] Antonietti, A., Ignazi, S. and Perego, P. (2000) Metacognitive knowledge about problem-solving methods. *British Journal of Educational Psychology*, **70**, pp. 1-16.
- [Arciszewski, 1988] Arciszewski, T. (1988) ARIZ 77: an Innovative Design Method. *Design Methods and Theories*, **22**, (2), pp. 796-820.
- [Arciszewski et al., 1995] Arciszewski, T., Michalski, R. S. and Wnek, J. (1995) Constructive Induction: The Key to Design Creativity. *Preprints "Computational Models of Creative Design"*, Third International Round-Table Conference, Heron Island, Australia, December 1995.
- [Austin et al., 2001] Austin, S., Steele, J., Macmillian, S., Kirby, P. and Spence, R. (2001) Mapping the conceptual design activity of interdisciplinary teams. *Design Studies*, **22**, pp. 211-232.
- [Ball et al., 2001] Ball, L. J., Lambell, N. J., Ormerod, T. C., Slavin, S. and Mariani, J. A. (2001) Representing design rationale to support innovative design reuse: a minimalist approach. *Automation in Construction*, **10**, pp. 663–674.
- [Boden, 1994] Boden, M. A. (1994) What is creativity? Dans *Dimensions of Creativity*, Massachussets Institute of Technology, pp. 75-117.
- [Bonnardel, 2000] Bonnardel, N. (2000) Towards understanding and supporting creativity in design: analogies in a constrained cognitive environment. *Knowledge-Based Systems*, **13**, pp. 505-513.
- [BTE, 1991] BTE (1991) *Maîtriser la conception par l'analyse de la valeur*, Paris.
-

-
- [Bultey, 2004] Bultey, A. (2004) *Modélisation de la TRIZ par les logiques de description*, Rapport de DEA In LICIA, INSA de Strasbourg, Strasbourg, France.
- [Caplat, 2002] Caplat, G. (2002) *Modélisation Cognitive et Résolution de Problèmes*, 2-88074-495-4, Lausanne.
- [Cavallucci, 1999] Cavallucci, D. (1999) *Contribution à la conception de nouveaux systèmes mécaniques par intégration méthodologique*, Rapport de thèse In Sciences Pour l'Ingénieur, Université Louis Pasteur, Strasbourg.
- [Cavallucci, 2001] Cavallucci, D. (2001) *La TRIZ, initiation et pratique*, LRPS, INSA Strasbourg, Strasbourg, 1996-2001.
- [Cavallucci et al., 2000] Cavallucci, D., Lutz, P. and Thiebaud, F. (2000) Intuitive Design Method (IDM): a new Framework for Design Method Integration. *The International Journal for Manufacturing Science & Production*, **3**, (2-4), pp. 95-101.
- [Champin, 2002] Champin, P.-A. (2002) *Modéliser l'expérience pour en assister la réutilisation. De la Conception Assistée par Ordinateur au web sémantique*, PhD In Université Claude Bernard, spécialité Informatique, Lyon I, Lyon.
- [Chan and Wu, 2002] Chan, L.-K. and Wu, M.-L. (2002) Quality function deployment: A literature review. *European Journal of Operational Research*, **143**, pp. 463-497.
- [Chandrasekaran and Johnson, 1993] Chandrasekaran, B. and Johnson, T. R. (1993) Generic tasks and task structures: History, critique and new directions. Dans *Second Generation Expert Systems*, (Ed, J.-M. David, J.-P. K., & R. Simmons (Ed.)) Springer-Verlag, Berlin, pp. 232-272.
- [Chosson, 1975] Chosson, J.-F. (1975) *L'entraînement mental*, Editions du Seuil, 2-02-002000-9, Paris.
- [Clancey, 1985] Clancey, W. J. (1985) Heuristic classification. *Artificial Intelligence*, **27**, (3), pp. 289-350.
-

-
- [Clancey, 1993] Clancey, W. J. (1993) The Knowledge Level Reinterpreted: Modeling Socio-Technical Systems. Dans *Knowledge Acquisition as Modeling*, (Ed, Eds, K. M. F. a. J. M. B.) Wiley, USA.
- [Cross, 1992] Cross, N. (1992) Research in Design Thinking. Dans *Research in Design Thinking*, (Eds, Cross, N., Dorst, K. and Roozenburg, N.) Delft University Press, Delft, Netherlands, pp. 3-10.
- [Crubézy and Musen, 2004] Crubézy, M. and Musen, M. A. (2004) Ontologies in Support of Problem Solving. Dans *Handbook on Ontologies*, (Ed, Studer, S. S. a. R.) Springer, pp. 321-342.
- [Crubleau, 2002] Crubleau, P. (2002) *L'identification des futures générations de produits industriels. Proposition d'une démarche utilisant les lois d'évolution de TRIZ*, In Génie Industriel, Institut des Sciences et Techniques d'Angers, Angers.
- [Darlington et al., 1998] Darlington, M., Potter, S., Culley, S. J. and Chawdry, P. K. (1998) *Cognitive theory as a guide to automating the configuration design process*, Kluwer Academic Publishers, Artificial Intelligence in Design '98, 0-7923-5159-2, Dordrecht.
- [De Araujo, 1996] De Araujo, C. S. (1996) Sharpening Understanding of Design Methods. *1st Annual International Conference on Industrial Engineering Applications and Practice*, Houston, Texas, USA, December 4-7.
- [de Bertrand de Beuvron et al., 1999] de Bertrand de Beuvron, F., Rousselot, F., Grathwohl, M., Rudloff, D. and Schlick, M. (1999) CICLOP. *International Workshop on Description Logics - DL'99*, Linköping, Sweden, July 30 - August 1.
- [De Bono, 1976] De Bono, E. (1976) *Teaching thinking*, Temple Smith, London.
-

-
- [De Carvalho and Back, 1999] De Carvalho, M. A. and Back, N. (1999) TRIZ Methodology and its Use in Systematic Engineering Design. *Congrès Brésilien d'Ingénierie Mécanique*, Aguas de Lindoia, Brazil.
- [Dewey, 1910] Dewey, J. (1910) *How we think*, Dover Publications, 0-486-29895-7, Boston, Boston: D. C. Heath, 1910.
- [Dieng, 1990] Dieng, R. (1990) *Méthodes et outils d'acquisition des connaissances*, INRIA (Institut National de Recherche en Informatique et en Automatique) - Unité de Sophia Antipolis, Le Chesnay, Novembre 1990.
- [Dikker et al., 1992] Dikker, F., Leemeijer, G. B., Wognum, P. M. and N.J.I., M. (1992) Towards computer-supported creative design based on Altshuller's theory of inventive problem solving. *Memoranda Informatica*, pp. 92-49.
- [Dortier, 2001] Dortier, J.-F. (2001) Histoire des Sciences Cognitives. *Sciences Humaines, Hors Série*, (35), pp. 6-14.
- [Dubois and Lutz, 2003a] Dubois, S. and Lutz, P. (2003a) Intégration de la TRIZ et de l'Axiomatic Design pour la Conception Inventive. *5^e Congrès International de Génie Industriel*, Québec, Canada, October 26-29.
- [Dubois and Lutz, 2003b] Dubois, S. and Lutz, P. (2003b) Representation of problems during the conceptual design: A roadmap from functional to physical domains. *12th International Conference on Management of Technology, IAMOT*, Nancy, France, May 13-15.
- [Duchamp, 1988] Duchamp, R. (1988) *La conception de produits nouveaux*, Hermes, Technologies de pointe, Paris.
- [Ellul, 1977] Ellul, J. (1977) *Le système technicien*, Calman-Lévy, Paris.
- [Eriksson, 2003a] Eriksson, H. (2003a) *JessTab*, Linköping, <http://www.ida.liu.se/~her/JessTab/>
-

-
- [Eriksson, 2003b] Eriksson, H. (2003b) JessTab: Using Jess together with Protégé. *Sixth International Protégé Workshop*, Manchester, England, 7-9 July.
- [Fensel and Motta, 1998] Fensel, D. and Motta, E. (1998) Structured Development of Problem Solving Methods. *11th Banff Knowledge Acquisition for Knowledge-Based System Workshop (KAW'98)*, Banff, Canada, April 1998.
- [Friedman-Hill, 2002] Friedman-Hill, E. (2002) *Jess*, 6.1, Livermore, CA, <http://herzberg.ca.sandia.gov/jess/>
- [Friedman-Hill, 2003] Friedman-Hill, E. (2003) *Jess in Action: Rule-Based Systems in Java*, Manning Publications Co., 1930110898.
- [Fuchs and Mille, 2000] Fuchs, B. and Mille, A. (2000) Une modélisation au niveau connaissance du raisonnement à partir de cas. *Journées francophones d'acquisition des connaissances - IC'2000*, Toulouse, France, 10-12 mai.
- [Galisson, 1996] Galisson, G. (1996) *Analyse Fonctionnelle*, Support de Cours In ENSAIS, Ecole Nationale Supérieure des Arts et Industries de Strasbourg, Strasbourg.
- [Gandon, 2002] Gandon, F. (2002) *Ontology Engineering: a Survey and a Return on Experience*, Institut de Recherche en Informatique et Automatique (INRIA Sophia Antipolis), mars 2002.
- [Gano, 2000] Gano, D. L. (2000) *Effective Problem Solving : a new way of thinking*, Apollo Associated Services, INC.
- [Garro, 2000] Garro, O. (2000) *Thématique de Recherche Conception Innovante et Distribuée*.
- [Gartiser et al., 2002] Gartiser, N., Kucharavy, D. and Lutz, P. (2002) Le Processus Convergent de la TRIZ: Une démarche Economiquement Efficace de Recherche de Solutions en Conception. *Colloque IPI*, Grenoble, France, 28-30 Janvier.
- [Gasanov, 1995] Gasanov, A. I. (1995) *Birth of the Invention*, Interpraks, in Russian, 5-85235-226-8, Moscow.
-

-
- [Gennari et al., 2003] Gennari, J. H., Musen, M. A., Ferguson, R. W., Grosso, W. E., Crubezy, M., Eriksson, H., Noy, N. F. and Tu, S. W. (2003) The evolution of Protege: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, pp. 89-123.
- [Gogu, 2001] Gogu, G. (2001) Méthodologie d'Innovation: la Résolution des Problèmes Créatifs. *Revue Française de Gestion Industrielle*, **19**, (3), pp. 35-62.
- [Goodman, 1978] Goodman, N. (1978) *Ways of worldmaking*, Hackett, Indianapolis.
- [Gordon, 1961] Gordon, W. J. (1961) *Synectics. The development of creative capacity*, Harper & Row, New York.
- [Grace et al., 2001] Grace, F., Slocum, M. S. and Clapp, D. (2001) A new TRIZ Practitioner's experience for Solving Problem using ARIZ 85C: Increasing a textile Kiss-Coat Operation speed. *TRIZ Journal*.
- [Guin, 1998] Guin, N. (1998) *Systèmes d'Aide à la Décision et à la Formation*, Rapport de Thèse In Informatique, Université Paris 6, Paris.
- [Gzara et al., 2003] Gzara, L., Rieu, D. and Tollenaere, M. (2003) Product information systems engineering: an approach for building product models by reuse of patterns. *Robotics and Computer Integrated Manufacturing*, (19), pp. 239-261.
- [Harutunian et al., 1996] Harutunian, V., Nordlund, M., Tate, D. and Suh, N. P. (1996) Decision Making and Software Tools for Product Development Based on Axiomatic Design Theory. *CIRP General Assembly*, Como, Italy, August 25-31.
- [Hatchuel and Weil, 2002] Hatchuel, A. and Weil, B. (2002) La théorie C-K. Fondements et usages d'une théorie unifiée de la conception. *International Conference "The Sciences of Design"*, Lyon, France, March 15-16.
-

-
- [Hauser and Clausing, 1988] Hauser, J. R. and Clausing, D. (1988) The House Of Quality. *Harvard Business Review*, **32**, (5), pp. 63-73.
- [Heidegger, 1953] Heidegger, M. (1953) La question de la technique. Dans *Essais et Conférences*, pp. 9-48.
- [Hsiao, 2002] Hsiao, S.-W. (2002) Concurrent Design Method for Developing a new Product. *International Journal of Industrial Ergonomics*, **29**, pp. 41-55.
- [Hutton, 1997] Hutton, D. (1997) *Quality Function Deployment*, www.dhutton.com.
- [Ideation, 2000a] Ideation, I. C. (2000a) *Improver*, 2.6.3,
- [Ideation, 2000b] Ideation, I. C. (2000b) *Innovation WorkBench*, 2.7.0,
- [Ideation, 1999] Ideation, R. G. (1999) *TRIZ in Progress*, Ideation International Inc., 1-928747-04-3, Southfield, Michigan.
- [Ideation, 2000c] Ideation, R. G. (2000c) *Knowledge Wizard*, 2.6.1,
- [Informatics, 2003] Informatics, S. M. (2003) *Protege-2000*, 1.9, <http://protege.stanford.edu>
- [Invention Machine, 1999] Invention Machine, C. (1999) *TechOptimizer Professional Edition*, 3.01 French, Boston, <http://www.invention-machine.com>
- [Ivanov and Bystritsky, 2000] Ivanov, G. and Bystritsky, A. (2000) *Formulation de problèmes de création*, Centre "TRIZ-INFO", Tcheliabinsk, LRPS, Strasbourg.
- [Khomenko, 2000a] Khomenko, N. (2000a) *OTSM-TRIZ: Contradiction Technology and Genrich Alsthuller's ARIZ-85-C*, Jonathan Livingston Project, Minsk, 1991-2000.
- [Khomenko, 2000b] Khomenko, N. (2000b) *OTSM-TRIZ: Introduction to Problem-Solving Technology*, "Jonathan Livingston" Project, Minsk, 1997-2000.
- [Khomenko, 2001] Khomenko, N. (2001) *Materials for Seminars: OTSM-TRIZ: Main Technologies of Problem Solving*, "Jonathan Livingston" Project, 1997-2001.
-

-
- [Khomenko and Sokol, 2000] Khomenko, N. and Sokol, A. (2000) *New Model and Methodology for Teaching OTSM-TRIZ*.
- [Killander and Sushkov, 1995] Killander, A. J. and Sushkov, V. V. (1995) Conflict-Oriented Model of Creative Design. *Third International Conference on Computational Models for Creative Design*, Heron Island, Queensland, Australia, December 3-7.
- [Kitamura and Mizoguchi, 2002] Kitamura, Y. and Mizoguchi, R. (2002) Ontology-based modeling of product functionality and use. Part 1: functional-knowledge modeling. *The Third International Seminar and Workshop Engineering Design in Integrated Product Development (EDIProD 2002)*, Lagow, Poland, October 10-12.
- [Kletke et al., 2001] Kletke, M. G., Mackay, J. M., Barr, S. H. and Jones, B. (2001) Creativity in the organization: the role of individual creative problem solving and computer support. *Int. J. Human-Computer Studies*, **55**, pp. 217-237.
- [Kolodner, 1993] Kolodner, J. (1993) *Case-based reasoning*, Morgan Kaufmann Publishers, Inc., 1-55860-237-2.
- [Kozyreva, 2002] Kozyreva, N. (2002) The Table for Diagnostic and Forecasting of the Development of Technical System. *TRIZ Future 2002*, Strasbourg, France.
- [Kucharavy, 2001a] Kucharavy, D. (2001a) *OTSM-TRIZ Technologies*, OTSM-TRIZ Technologies Center, Minsk, Belarus. Laboratoire de Recherche en Productique de Strasbourg (ENSAIS), France, Strasbourg, November 2001 - March 2001.
- [Kucharavy, 2001b] Kucharavy, D. (2001b) *TRIZ: Techniques*, OTSM-TRIZ Technologies Center, Minsk, Belarus. Laboratoire de Recherche en Productique de Strasbourg (ENSAIS), France., Strasbourg, November-December 2001.
- [Kucharavy, 2002] Kucharavy, D. (2002) *Russian-English-French OTSM-TRIZ Dictionary of Conformities*, OTSM-TRIZ Technologies Center, Minsk, Belarus, 1998-2002.
-

-
- [Kurr, 1998] Kurr, T. T. (1998) *Synthesis of the Principle-Based and Other Product Development Approaches with Emphasis on Concept Generation and Evaluation*, Lehrstuhl für Fertigungsmesstechnik und Qualitätsmanagement In Laboratorium für Werkzeugmaschinen und Betriebslehre, Rheinisch-Westfälischen Technischen Hochschule, Aachen.
- [Lawson et al., 2003] Lawson, B., Bassanino, M., Phiri, M. and Worthington, J. (2003) Intentions, practices and aspirations: Understanding learning in design. *Design Studies*, **24**, pp. 327-339.
- [Le Masson, 2002] Le Masson, P. (2002) *Management de l'Innovation et Ingénierie de la Conception*, CGS, Ecole des Mines, Paris.
- [Leon-Rovira and Aguayo, 2001] Leon-Rovira, N. and Aguayo, H. (2001) *A New Model of the Conceptual Design Process Using QFD/FA/TRIZ*, 1/2/01, www.ideationtriz.com.
- [Leroi-Gourhan, 1971] Leroi-Gourhan, A. (1971) *L'homme et la matière*, Sciences d'aujourd'hui, Paris.
- [Linde and Herr, 2002] Linde, H. J. and Herr, G. H. (2002) Innowis and Wois. *TRIZ Future 2002*, Strasbourg, France.
- [Liu and Bligh, 2003] Liu, Y. C. and Bligh, T. (2003) Towards an ideal approach for concept generation. *Design Studies*, **24**, (4), pp. 341-355.
- [Maier, 1963] Maier, N. R. F. (1963) Locating the problem. Dans *Problem-solving discussions and conferences*, Mc Graw-Hill Book Company, Inc., pp. 49-73.
- [Malmquist et al., 1996] Malmquist, J., Axelsson, R. and Johansson, M. (1996) A Comparative Analysis of the Theory of Inventive Problem Solving and the Systematic Approach of Pahl and Beitz. *DSTC*, Irvine.
- [Mann, 1999] Mann, D. (1999) Axiomatic Design and TRIZ: Compabilities and Contradictions. *TRIZ Journal*.
- [Marconi, 1998] Marconi, J. (1998) ARIZ: the algorithm for Inventive Problem Solving. *TRIZ Journal*.
-

-
- [Marcus, 1988] Marcus, S. (1988) *Automating Knowledge Acquisition for Experts Systems*, Kluwer Academic Publisher, Boston.
- [Maslow, 1954] Maslow, A. H. (1954) *Motivation and Personality*, Harper and Row, New York.
- [Mathieu, 1986] Mathieu, J. (1986) Psychologie cognitive et intelligence artificielle. Résolution de problème et acquisition de connaissances. Dans *Psychologie, Intelligence Artificielle et Automatique*, (Ed, Pierre Mardaga, E.) Bruxelles, Belgique, pp. 19-36.
- [Matta, 1995] Matta, N. (1995) *Méthodes de Résolution de Problèmes : leur explicitation et leur représentation dans MACAO-II*, Rapport de Thèse In Informatique, Université Paul Sabatier, Toulouse.
- [Mayer, 2002] Mayer, R. E. (2002) A taxonomy for computer-based assessment of problem-solving. *Computers in Human Behavior*, **18**, pp. 623-632.
- [McDermott, 1990] McDermott, J. (1990) The world would be a better place if Non-Programmers could program. Dans *Knowledge Acquisition: Selected Research and Commentary*, (Ed, Editor, S. M.) Kluwer Academic Publishers, Netherlands.
- [Mili et al., 2001] Mili, F., Shen, W., Martinez, I., Noel, P., Ram, M. and Zouras, E. (2001) Knowledge modelling for design decisions. *Artificial Intelligence in Engineering*, **15**, (2), pp. 153-164.
- [Motta and Zdrahal, 1998] Motta, E. and Zdrahal, Z. (1998) A library of problem-solving components based on the integration of the search paradigm with task and method ontologies. *International Journal of Human-Computer Studies*, **49**, (4), pp. 437-470.
- [Musen et al., 2000] Musen, M. A., Ferguson, R. W., Grosso, W. E., Noy, N. F., Crubezy, M. and Gennari, J. H. (2000) Component-Based Support for Building Knowledge-Acquisition Systems. *Conference on Intelligent Information Processing (IIP 2000)*
-

-
- of the International Federation for Information Processing World Computer Congress (WCC 2000)*, Beijing.
- [NF X 50 100] *Analyse Fonctionnelle – Caractéristiques fondamentales*, Editée par Association Française de NORmalisation (AFNOR), version de 1996
- [NF X 50 127] *Recommandations pour obtenir et assurer la qualité en conception*, Editée par Association Française de NORmalisation (AFNOR), version de Janvier 1988
- [NF X 50 150] *Analyse de la Valeur, Analyse Fonctionnelle – Vocabulaire*, Editée par Association Française de NORmalisation (AFNOR), version de 1990
- [NF X 50 150-1] *Vocabulaire du management, de la Valeur, de l'Analyse de la Valeur et de l'analyse fonctionnelle*, Editée par Association Française de NORmalisation (AFNOR), version de 1996
- [NF X 50 152] *Analyse de la Valeur, Caractéristiques Fondamentales*, Editée par Association Française de NORmalisation (AFNOR), version de 1990
- [Osborn, 1953] Osborn, A. (1953) *Applied Imagination*, Scribner, New York.
- [Oxman, 1994] Oxman, R. (1994) Precedents in Design: a computational model for the organization of precedent knowledge. *Design Studies*, **15**, (2), pp. 141-157.
- [Pahl and Beitz, 1988] Pahl, G. and Beitz, W. (1988) *Engineering design: a systematic approach*, Springer Verlag, 0-387-50442-7, New York.
- [Peltola et al., 2003] Peltola, P., Leppimäki, S., Meristö, T. and Bergmann, J.-P. (2003) The Methods of Futures Research in the Product Concept Generation Process. *12th International Conference on Management of Technology, IAMOT*, Nancy, France, May 13-15.

-
- [Perner, 2002] Perner, P. (2002) Are case-based reasoning and dissimilarity-based classification two sides of the same coin? *Engineering Applications of Artificial Intelligence*, **15**, pp. 193 –203.
- [Perrin, 2001] Perrin, J. (2001) *Concevoir l'innovation industrielle. Méthodologie de conception de l'innovation*, 2-271-05822-8, Paris.
- [Pinyayev] Pinyayev, A. M. *Transition from the administrative contradiction to the technical one in the analysis of inventive problems*, Traduit du russe par Julie Stien, LRPS, Strasbourg.
- [Prudhomme, 1999] Prudhomme, G. (1999) *Le processus de conception de systèmes mécaniques et son enseignement - La transposition didactique comme outil d'analyse épistémologique*, Thèse de doctorat In Université Joseph Fourier, Grenoble.
- [Rechenmann, 1995] Rechenmann, F. (1995) Les catégories de connaissance et leur modélisation. *2^{ndes} rencontres Théo Quant sur « décision spatiale»*, Besançon, France, 4-5 octobre.
- [Richards and Simoff, 2001] Richards, D. and Simoff, S. J. (2001) Design ontology in context - a situated cognition approach to conceptual modelling. *Artificial Intelligence in Engineering*, **15**, pp. 121-136.
- [Roucoules et al., 2003] Roucoules, L., Lutz, P. and Gartsier, N. (2003) Several Dimension for Characterizing Innovation: a Technical and Economical Approach. *International Conference on Engineering Design (ICED'03)*, Stockholm, Sweden, 19-21 August, 2003.
- [Rousselot, 2002] Rousselot, F. (2002) *LIKES (LInguistic and Knowlege Engineer-ing Station) : outil de traitement de corpus et d'aide à la construction d'ontologies*, www-ensais.u-strasbg.fr/liia/likes/likes.htm
- [Rudolph, 1996] Rudolph, S. (1996) Upper and Lower Limits for "The Principles of Design". *Research in Engineering Design*, **8**, pp. 207-216.
-

-
- [Salamatov, 1996] Salamatov, Y. P. (1996) *Les Lois d'évolution des Systèmes Techniques (Fondements de la Théorie d'Évolution des Systèmes Techniques)*, Version traduite du russe, LRPS-INSA Strasbourg, 2000-2002.
- [Salamatov, 1999] Salamatov, Y. P. (1999) *TRIZ: The Right Solution at the Right Time*, Insytec B. V., 90-804680-1-0, Hattem, The Netherlands.
- [Savransky, 2000] Savransky, S. D. (2000) *Engineering of Creativity: Introduction to TRIZ Methodology of Inventive Problem Solving*, CRC Press LLC, 0-8493-2255-3, Boca Raton, Florida.
- [Schlueter, 2001] Schlueter, M. (2001) QFD by TRIZ. *TRIZCON2001, Third Annual Altshuller Institute for TRIZ Studies International Conference*, Woodland Hills, CA, March 25-27.
- [Schreiber, 2000] Schreiber, A. T. (2000) *Problem-Solving Methods for knowledge Systems: an Introduction*, Social Science Informatics, University of Amsterdam, Amsterdam, 2000-2001.
- [Simon, 1987] Simon, H. A. (1987) Problem Forming, Problem Finding, and Problem Solving. *1st International Congress on Planning and Design Theory*, Boston, USA.
- [Simon, 1991] Simon, H. A. (1991) *Sciences des Systèmes, Sciences de l'Artificiel*, Dunod, Paris, (1969) - *The Sciences of the Artificial*, Cambridge, The MIT Press.
- [Sivaloganathan et al., 2001] Sivaloganathan, S., Andrews, P. T. J. and Shahin, T. M. M. (2001) Design Function Deployment: a tutorial introduction. *Journal of Engineering Design*, **12**, (1), pp. 59-74.
- [Slocum, 2002] Slocum, M. S. (2002) Measurement and Detection Standards from the Theory of Inventive Problem Solving. *Annual American Physical Society March Meeting*, Indianapolis, Indiana, March 18 - 22.
-

-
- [Soderlin, 2003] Soderlin, P. (2003) Thoughts on ARIZ. Do we need to redesign the ARIZ 2000? *TRIZ Journal*.
- [Souckov, 1998] Souckov, V. V. (1998) *Knowledge-Based Support for Innovative Design*, In Univesity of Twente, Twente.
- [Stefik, 1995] Stefik, M. (1995) *Introduction to Knowledge Systems*, Morgan Kaufmann Publishers, Inc., 1-55860-166-X, San Francisco.
- [STEM-Net, 1996] STEM-Net (1996) *La conception en tant que moyen de résoudre les problèmes*, Ministère de l'éducation de la province de Terre-Neuve et du Labrador.
- [Studer et al., 1998] Studer, R., Benjamins, V. R. and Fensel, D. (1998) Knowledge Engineering: Principles and methods. *Data & Knowledge Engineering*, **25**, pp. 161-197.
- [Suh, 1990] Suh, N. P. (1990) *The Principles of Design*, Oxford University Press, 0-19-504345-6, New York.
- [Suh, 2001] Suh, N. P. (2001) *Axiomatic Design: Advances and Applications*, Oxford University Press, 0-19-513466-4, New York.
- [Suh et al., 1978] Suh, N. P., Kim, S. H., Wilson, D. R., Cook, N. N., Lapidot, N. and Turkowich, B. (1978) Optimization of manufacturing systems trough axiomatics. *Annals of the CIRP*, **27**, (1), pp. 383-388.
- [Sullivan, 1986] Sullivan, L. P. (1986) Quality function deployment. *Quality Progress*, **19**, (6), pp. 39-50.
- [Tate, 1999] Tate, D. (1999) *A Roadmap for Decomposition: Activities, Theories, and Tools for System Design*, Doctor of Philosophy In Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA.
- [Tate and Nordlund, 1995] Tate, D. and Nordlund, M. (1995) Synergies Between American and European Approaches to Design. *First World*
-

-
- Conference on Integrated Design and Process Technology (IDPT)*, Austin, TX, December 7-9.
- [Tate and Nordlund, 1996] Tate, D. and Nordlund, M. (1996) A Design Process Roadmap as a General Tool for Structuring and Supporting Design Activities. *Second World Conference on Integrated Design and Process Technology (IDPT-Vol. 3)*, Austin, TX, Dec. 1-4.
- [Temponi et al., 1999] Temponi, C., Yen, J. and Tiao, W. A. (1999) Theory and Methodology. House of quality: A fuzzy logic-based requirements analysis. *European Journal of Operational Research*, **117**, pp. 340-354.
- [Terninko et al., 2000] Terninko, J., Domb, E. and Miller, J. (2000) The Seventy-six Standard Solutions, with Examples. Section One. *TRIZ Journal*.
- [Terninko et al., 1998] Terninko, J., Zusman, A. and Zlotin, B. (1998) *Systematic Innovation – An Introduction to TRIZ*, St. Lucie Press, 1-57444-111-6.
- [Trichet, 1998] Trichet, F. (1998) *DSTM: un environnement de modélisation et d'opérationnalisation de la démarche de résolution de problèmes d'un Système à Base de Connaissances*, Rapport de Thèse In Informatique, Faculté des Sciences et des Techniques, Université de Nantes, Nantes.
- [TriSolver, 2002] TriSolver, C. (2002) *TriSolver*, 2.1,
- [Tu et al., 1995] Tu, S. W., Eriksson, H., Gennari, J. H., Shahar, Y. and Musen, M. A. (1995) Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: application of PROTEGE-II to protocol-based decision support. *Artificial Intelligence in Medicine*, **7**, pp. 257-289.
- [Ullman, 2001] Ullman, D. G. (2001) Robust decision-making for engineering design. *Journal of Engineering Design*, **12**, (1), pp. 3-13.
-

-
- [Ulrich and Eppinger, 1995] Ulrich, K. T. and Eppinger, S. D. (1995) *Product Design and Development*, New York.
- [Valente et al., 1998] Valente, A., Breuker, J. and Van de Velde, W. (1998) The CommonKADS library in perspective. *Int. J. Human-Computer Studies*, **49**, pp. 391-416.
- [Vidal, 1999] Vidal, F. (1999) Méthodes de créativité. Dans *Techniques de l'Ingénieur, traité Génie Industriel*, Vol. A 5 210 pp. 1-16.
- [Visser, 2004] Visser, W. (2004) *Dynamic Aspects of Design Cognition: Elements for a Cognitive Model of Design*, Institut National de Recherche en Informatique et en Automatique, Rocquencourt, France, Mars 2004.
- [Yang and Zhang, 2000] Yang, K. and Zhang, H. (2000) A Comparison of TRIZ and Axiomatic Design. *TRIZ Journal*.
- [Yannou, 2001] Yannou, B. (2001) *Préconception de Produits*, Mémoire d'Habilitation à Diriger des Recherches In Mécanique, Institut National Polytechnique de Grenoble, Grenoble.
- [Yannou and Hajsalem, 2001] Yannou, B. and Hajsalem, S. (2001) Comparaison des apports de l'Analyse de la Valeur et de la méthode SPEC sur un cas industriel. *PRIMECA'2001: Septième Colloque sur la Conception Mécanique Intégrée*, La Plagne, France.
- [Yannou et al., 2002] Yannou, B., Hajsalem, S. and Limayem, F. (2002) Comparaison de la méthode SPEC et de l'Analyse de la Valeur pour l'aide à la conception préliminaire de produits. *Mécanique & Industries*, **3**, pp. 189-199.
- [Zakharov, 1993] Zakharov, A. N. (1993) *Construction du système LDST (Lois d'évolution des systèmes techniques)*, traduit du russe par Julie Stien, LRPS, Strasbourg.
- [Zlotin and Zusman, 1991] Zlotin, B. and Zusman, A. (1991) *Problems of ARIZ Enhancement*, 1/2/01.
- [Zlotin and Zusman, 1998] Zlotin, B. and Zusman, A. (1998) *ARIZ on the move*, http://www.ideationtriz.com/paper_ARIZontheMove.htm.
-

[Zlotin et al., 2000]

Zlotin, B., Zusman, A. and Kaplan, L. (2000) *Containment Ring Problem. A comparative case study using the Contradiction Table, Improver software, and Innovation WorkBench (IWB) software*, 3\20\01.

ANNEXES

Annexe-I	Liste des 76 standards de résolution des problèmes d'innovation.....	193
Annexe-II	ARIZ-71 extraits, selon [Altshuller, 1973].....	217
Annexe-III	ARIZ-77, selon [Arciszewski, 1988].....	221
Annexe-IV	Algorithm of Inventive Problem Solving : ARIZ-85C.....	223
Annexe-V	Définition de la représentation centrée-objet.....	240
Annexe-VI	Ecriture du modèle de connaissances en logique de description.....	260
Annexe-VII	Règles définies dans JESS.....	270
Annexe-VIII	Liste des questions.....	291
Annexe-IX	Liste des fonctions définies dans Jess.....	306
Annexe-X	Première série d'expérimentations.....	326
Annexe-XI	Seconde série d'expérimentations.....	336

ANNEXE-I. LISTE DES 76 STANDARDS DE RESOLUTION DES PROBLEMES D'INNOVATION

Classe 1 : Construction et destruction des systèmes vépoles

Sous-classe 1.1 : Synthèse des vépoles

Descriptif :

L'idée principale de cette sous-classe est exprimée clairement par le standard 1.1.1: pour synthétiser un Système Technique (ST) opérationnel, il faut, dans le cas le plus simple, passer du non-vépole au vépole. La construction du vépole se heurte souvent aux difficultés conditionnées par différentes contraintes à l'introduction de substances et de champs. Les standards 1.1.2 à 1.1.8 montrent des détours typiques dans ces cas.

Standard 1.1.1 : Construction du vépole

Si un objet résiste aux changements nécessaires et si les conditions du problème ne contiennent pas de contraintes à l'introduction de substances et de champs, alors on résout le problème par la synthèse du vépole en introduisant les éléments qui manquent.

Remarque :

On est souvent amené à créer des vépoles dans les problèmes où nous devons effectuer des opérations avec des objets fins, fragiles ou faciles à déformer. Pendant la réalisation des opérations, on rassemble l'objet avec une substance qui le rend dur et solide, puis on élimine cette substance par dissolution, par évaporation, etc.

Standard 1.1.2 : Vépole complexe intérieur

Si un vépole résiste aux changements nécessaires et si les conditions du problème ne contiennent pas de contraintes à l'introduction d'additifs dans les substances, alors on résout le problème par la transition (définitive ou provisoire) vers un vépole complexe intérieur en introduisant dans S1 et S2 des additifs qui augmentent la contrôlabilité ou qui donnent au vépole les propriétés nécessaires.

Remarque :

Souvent les conditions d'un problème comportent deux substances qui interagissent mal ou n'interagissent pas du tout avec le champ. Le vépole semble exister (tous les trois éléments sont présents) et en même temps ne pas exister parce qu'il est incomplet. Les détours les plus simples, dans ce cas, consistent à introduire des additif, intérieurs (dans une des substances) et extérieurs (sur une des substances). Ces vépoles sont nommés complexes (standards 1.1.2 et 1.1.3).

Parfois la même résolution, selon la manière dont le problème est posé, peut être écrite comme la construction du vépole ou comme la construction du vépole complexe.

Standard 1.1.3 : Vépole complexe extérieur

Si un vépole résiste aux changements nécessaires et si les conditions du problème contiennent des contraintes à l'introduction d'additifs dans les substances S1 et S2, alors on résout le problème par la transition (définitive ou provisoire) vers le vépole complexe extérieur en ajoutant à S1 ou à S2 une substance étrangère S3 qui augmente la contrôlabilité ou donne au vépole les propriétés nécessaires.

Standard 1.1.4 : Vépole basé sur l'environnement

Si un vépole résiste aux changements nécessaires et si les conditions du problème contiennent des contraintes à l'introduction ou à l'ajout de substances, alors on résout le problème par l'achèvement du vépole en utilisant l'environnement en tant que substance à introduire.

Standard 1.1.5 : Vépole basé sur l'environnement avec les additifs

Si l'environnement ne contient pas de substances nécessaires pour construire un vépole selon le standard 1.1.4, alors on peut obtenir une telle substance en remplaçant l'environnement, en le décomposant ou en y introduisant des additifs.

Standard 1.1.6 : Régime minimal

Si l'on a besoin du régime minimal (dosé, optimal) d'action et si, selon les conditions du problème, il est difficile ou impossible de l'assurer, alors il faut utiliser le régime maximal et

enlever l'excès. Dans ce cas, on enlève l'excès de champ par la substance et l'excès de substance par le champ.

Standard 1.1.7 : Régime maximal

S'il faut assurer le régime maximal d'action sur la substance et si cela est inadmissible pour telle ou telle raison, alors il faut garder l'action maximale, mais il faut la diriger sur une autre substance qui est liée avec la première.

Standard 1.1.8 : Régime maximal sélectif

Si l'on a besoin du régime maximal sélectif (régime maximal dans certaines zones tout en gardant le régime minimal dans d'autres zones), alors le champ doit être soit maximal soit minimal:

- maximal. Dans ce cas, on introduit une substance protectrice dans les endroits où l'action minimale est nécessaire (sous-groupe 1.1.8.1);
- minimal. Dans les endroits où l'action maximale est nécessaire, on introduit une substance qui crée un champ local, par exemple, des éléments thermogènes pour une action thermique ou des éléments explosifs pour une action mécanique (sous-groupe 1.1.8.2).

Sous-classe 1.2 : Destruction des vépoles

Descriptif :

La sous-classe 1.2 comporte les standards de la destruction des vépoles et de l'élimination ou de la neutralisation des liens nuisibles. L'idée la plus forte de cette sous-classe est la mobilisation des éléments nécessaires en utilisant les ressources des champs et des substances. Dans la pratique, le standard 1.2.2 est particulièrement important. Selon ce standard, la substance déjà contenue dans le système, mais modifiée, remplit les fonctions d'une nouvelle substance.

Standard 1.2.1 : Elimination du lien nuisible par l'introduction de S3

Si dans un vépole les actions utile et nuisible se produisent en même temps entre deux substances et s'il n'est pas obligatoire de garder le contact direct entre elles, alors on résout le

problème en introduisant entre ces substances une troisième substance étrangère gratuite ou assez bon marché.

Standard 1.2.2 : Elimination du lien nuisible par l'introduction de S1 et S2 modifiées

Si dans un vépole les actions utiles et nuisibles se produisent en même temps entre deux substances et s'il n'est pas obligatoire de garder le contact direct entre elles ; et que l'utilisation de substances étrangères est interdite ou inutile, alors on résout le problème en introduisant une troisième substance qui est une modification des deux premières (voir la formule vépole du standard 1.2.1).

Remarque :

On peut introduire la substance S3 dans le système de l'extérieur sous forme prête à utiliser ou bien on peut l'extraire (par l'action de C1 et C2) des substances présentes. Par exemple, S3 peut être le "vide", les bulles d'air, la mousse, etc.

Standard 1.2.3 : "Détournement" de l'action nuisible

S'il faut éliminer une action nuisible du champ sur la substance, alors on peut résoudre le problème en introduisant un deuxième élément qui attire l'action nuisible du champ sur lui.

Standard 1.2.4 : Résistance aux liens nuisibles à l'aide de C2

Si dans un vépole les actions utile et nuisible se produisent en même temps entre deux substances et si le contact direct entre elles (à la différence des standards 1.2.1 et 1.2.2) doit être gardé, alors on résout le problème par la transition vers le double vépole dans lequel l'action utile reste au champ C1 et la neutralisation de l'action nuisible (ou la transformation de l'action nuisible en deuxième action utile) est effectuée par C2.

Standard 1.2.5 : "Débranchement" des liens magnétiques

Si le vépole contenant le champ magnétique doit être détruit, alors on peut résoudre le problème à l'aide des effets physiques qui "débranchent" les propriétés ferromagnétiques des

substances. Par exemple, en démagnétisant la substance à la suite d'un coup ou d'un échauffement au-dessus du point de Curie.

Classe 2 : Evolution des systèmes vépoles

Sous-classe 2.1 : Transition vers les vépoles complexes

Descriptif :

Tout d'abord, on peut accroître l'efficacité des vépoles par la transition des simples vépoles aux vépoles complexes (en chaîne et doubles). La complexité, dans ce cas, est relativement peu importante. Cependant, la transition assure l'apparition de nouvelles propriétés et le renforcement des propriétés déjà existantes. En premier lieu, elle renforce la contrôlabilité du système.

Standard 2.1.1 : Vépoles en chaîne

Pour accroître l'efficacité d'un système vépole, on résout le problème en transformant une des parties du vépole en un vépole indépendamment contrôlable et en créant ainsi le vépole en chaîne. A son tour, la substance introduite peut être développée en vépole.

Remarques :

1. *S'il y a dans le ST un objet qui se déplace ou qui doit se déplacer autour d'un axe sous l'action de la force d'attraction et s'il faut contrôler le mouvement de cet objet, alors on résout le problème en introduisant dans l'objet une substance qui se déplace d'une manière contrôlée à l'intérieur de l'objet et qui déplace par son mouvement le centre d'attraction du système.*

2. *On peut créer un vépole en chaîne en développant le lien entre les substances en un vépole. En ce cas, une nouvelle substance et un nouveau champ sont à introduire entre les substances du vépole donné.*

Standard 2.1.2 : Doubles vépoles

Si un vépole est difficilement contrôlable et s'il faut augmenter son efficacité et que la substitution des éléments de ce vépole est inadmissible, alors on résout le problème par la construction du double vépole en introduisant un deuxième champ facilement contrôlable.

Sous-classe 2.2 : Renforcement des vépoles

Descriptif :

L'idée générale des standards de cette sous-classe consiste à accroître l'efficacité des vépoles (simples et complexes) sans introduire de nouveaux champs et substances. On peut y parvenir en utilisant d'une manière renforcée, les ressources des substances et des champs.

Standard 2.2.1 : Transition vers les champs mieux contrôlables

On peut accroître l'efficacité d'un système vépole en remplaçant le champ de travail non-contrôlable (ou difficilement contrôlable) par un champ contrôlable (ou facilement contrôlable). Par exemple, en remplaçant le champ de gravitation par un champ mécanique, le champ mécanique par un champ électrique, etc.

Standard 2.2.2 : Concassage de S2

On peut accroître l'efficacité d'un système vépole en augmentant le degré de dispersion (concassage) de la substance qui joue le rôle de l'instrument.

Remarque :

Le standard 2.2.2 illustre une des lois principales d'évolution des ST, notamment la tendance à concasser l'instrument ou une de ses parties qui interagit directement avec l'objet.

Standard 2.2.3 : Transition vers les substances capillairo-poreuses

Un cas particulier du concassage de substance est la transition des substances compactes vers les substances capillairo-poreuses. On effectue cette transition selon le schéma suivant : "substance compacte – substance compacte avec une cavité – substance compacte avec plusieurs cavités (substance perforée) – substance capillairo-poreuse – substance capillairo-poreuse avec une certaine structure (et les dimensions) de pores".

Au fur et à mesure que la substance évolue suivant ce schéma, la possibilité de placer une substance liquide dans ses cavités (les pores) et d'utiliser les effets physiques augmente.

Standard 2.2.4 : Dynamisation

On peut accroître l'efficacité d'un système vépole en augmentant le degré de sa dynamisation, c'est-à-dire sa transition vers la structure plus souple et rapidement modifiable.

Remarques :

1. La dynamisation de S2 commence le plus souvent par la division de S2 en deux parties rassemblées à l'aide de charnières. Ensuite la dynamisation suit ce schéma : "une charnière – plusieurs charnières – S2 souple".

2. Dans le cas le plus simple, la dynamisation d'un champ C s'effectue par la transition de l'action permanente de champ (ou de C avec S2) vers l'action par impulsion.

3. Une dynamisation efficace du système peut être réalisée par les transitions de phase du premier degré (par exemple, congélation d'eau ou décongélation de glace) ou du second degré (par exemple, l'effet "mémoire de forme").

Standard 2.2.5 : Structuration des champs

On peut accroître l'efficacité d'un système vépole en passant des champs homogènes ou des champs ayant une structure désordonnée vers les champs hétérogènes ou les champs ayant une certaine structure dans l'espace (permanente ou variable).

Si l'on doit structurer dans l'espace une substance qui fait partie du vépole (ou qui est susceptible d'en faire partie), alors on doit le faire dans un champ qui a une structure correspondante à celle exigée.

S'il faut répartir à nouveau l'énergie du champ, par exemple, pour la concentration ou, au contraire, s'il faut créer des zones où l'action du champ n'apparaît pas, il est conseillé de passer à l'utilisation des ondes stationnaires.

Remarque :

Le standard 2.2.4 est souvent utilisé en combinaison avec le standard 1.2.5 ("débranchement" des liens magnétiques").

Standard 2.2.6 : Structuration des substances

On peut accroître l'efficacité d'un système vépole en passant des substances homogènes ou des substances ayant une structure désordonnée vers les substances hétérogènes ou les substances ayant une certaine structure dans l'espace (permanente ou temporaire).

Sous-classe 2.3 : Renforcement par l'accord des rythmes

Descriptif :

La sous-classe 2.3 comprend les standards de renforcement des vépoles par les moyens particulièrement économiques. Au lieu d'introduire ou de changer considérablement les substances et les champs, les standards de la sous-classe 2.3 envisagent des modifications uniquement quantitatifs, à savoir des modifications de fréquences, dimensions, masse. Ainsi, on obtient un nouvel effet important avec un minimum de changements dans le système.

Standard 2.3.1 : Accord des rythmes de C et S1 (ou S2)

Dans les systèmes vépoles la fréquence de l'action du champ doit être accordée (ou désaccordée sciemment) avec la fréquence propre à l'objet (ou à l'instrument).

Standard 2.3.2 : Accord des rythmes de C1 et C2

Dans les systèmes vépoles complexes les fréquences des champs utilisés doivent être accordées (ou désaccordées sciemment).

Standard 2.3.3 : Accord des actions incompatibles ou étant auparavant indépendantes

Si deux actions, comme le changement et la mesure, sont incompatibles, alors on effectue une action dans les pauses de l'autre. En général, les pauses d'une action doivent être remplies d'une autre action utile.

Sous-classe 2.4 : Fépoles (vépoles complexes renforcés)

Descriptif :

Le renforcement peut suivre quelques voies standards à la fois. Les fépoles (c'est-à-dire les vépoles avec une substance ferromagnétique concassée et un champ magnétique) se laissent renforcer le plus.

Standard 2.4.1 : "Protofépoles"

On peut accroître l'efficacité d'un système vépole en utilisant une Substance Ferromagnétique (SF) et un Champ Magnétique (CM).

Remarques :

Il s'agit d'utiliser une substance ferromagnétique SF non-concassée (c'est-à-dire des "protofépoles", "mi-fépoles") qui constitue une structure sur le chemin vers les fépoles.

Ce standard peut être appliqué non seulement aux vépoles simples mais aussi aux vépoles complexes et aux vépoles comprenant l'environnement.

Standard 2.4.2 : Fépoles

Pour accroître l'efficacité du contrôle d'un système, il faut passer du vépole ou du "protofépole" au fépole, après avoir remplacé (ou ajouté) une des substances, comme copeaux, granules, grains, etc., par des particules ferromagnétiques SF en utilisant le champ magnétique ou électromagnétique. Plus le degré de concassage des particules ferromagnétiques est élevé, plus le contrôle est efficace. C'est pourquoi l'évolution des fépoles suit ce schéma : "granules – poudre – particules ferromagnétiques concassées". L'efficacité s'accroît également lorsque le degré du concassage de la substance, dans laquelle sont introduites des particules ferromagnétiques, augmente. Donc, l'évolution suit la voie : "substance solide – grains – poudre – liquide".

Remarques :

On peut considérer la transition vers les fépoles comme l'application simultanée des standards 2.2.1 (concassage de substance) et 2.4.1 (introduction de substance ferromagnétique et de champ magnétique).

Après être transformé en fépole, le système vépole reprend le cycle d'évolution des vépoles, mais à un autre niveau, car les fépoles se distinguent par la haute contrôlabilité et par l'efficacité. Tous les standards qui font partie de la sous-classe 2.4 peuvent être considérés comme une sorte d' "isotopes" de la suite ordinaire des standards (2.1 à 2.3).

La distinction des fépoles en une sous-classe séparée est justifiée par leur importance exclusive dans la pratique (en tout cas à cette étape d'évolution du système des standards).

En outre, la "suite fépole" est pertinente en tant qu'un instrument fin pour l'étude de la "suite vépole", plus rude, et pour la prédiction de son évolution.

Standard 2.4.3 : Liquide magnétique

On peut accroître l'efficacité des fépoles en passant à l'utilisation des liquides magnétiques, c'est-à-dire des particules ferromagnétiques colloïdales dispersées dans le kérosène, le silicone ou dans l'eau.

On peut considérer le standard 2.4.3 comme un cas extrême de développement du standard 2.4.2.

Standard 2.4.4 : Utilisation des structures capillairo-poreuses dans les fépoles

On peut accroître l'efficacité des fépoles en utilisant la structure capillairo-poreuse propre à un grand nombre de systèmes fépoles.

Standard 2.4.5 : Fépoles complexes

S'il faut accroître l'efficacité de contrôle d'un système par la transition vers les fépoles et si le remplacement de la substance par les particules ferromagnétiques est inadmissible, alors on effectue la transition par la construction du fépole complexe intérieur ou extérieur en introduisant des additifs dans l'une des substances.

Standard 2.4.6 : Fépoles basés sur l'environnement

S'il faut accroître l'efficacité de contrôle d'un système par la transition vers le fépole, mais que le remplacement de la substance par les particules ferromagnétiques (ou l'introduction d'additifs dans la substance) est inadmissible, alors il faut introduire des particules ferromagnétiques dans l'environnement et modifier ses paramètres en agissant par le champ magnétique. Par conséquent, on peut contrôler le système qui se trouve dans cet environnement (standard 2.4.3).

Si dans le système on utilise des flotteurs ou que l'une des parties du système est un flotteur, alors il faut introduire dans le liquide des particules ferromagnétiques et contrôler la viscosité du liquide. On peut aussi effectuer le contrôle en faisant passer le courant électrique à travers le liquide et en agissant par le champ électromagnétique.

On peut également utiliser des liquides électrorhéologiques commandées par un champ électrique, tel que l'environnement.

Standard 2.4.7 : Utilisation des effets physiques

On peut accroître l'efficacité d'un système vépole en utilisant les effets physiques.

Standard 2.4.8 : Dynamisation

On peut accroître l'efficacité d'un système vépole par la dynamisation, c'est-à-dire par la transition vers une structure souple et modifiable du système.

Standard 2.4.9 : Structuration

On peut accroître l'efficacité d'un système vépole en passant des champs homogènes ou des champs ayant une structure désordonnée aux champs hétérogènes ou aux champs ayant une structure ordonnée dans l'espace (permanente ou temporaire).

Si l'on doit structurer dans l'espace une substance qui fait partie du fépole (ou qui est susceptible d'en faire partie), alors il faut le faire dans un champ qui possède la structure correspondante à celle exigée.

Standard 2.4.10 : Accord des rythmes dans les fépoles

On peut accroître l'efficacité d'un système "protofépole" ou fépole en accordant les rythmes des éléments de ce système.

Standard 2.4.11 : Epoles

Si l'introduction des ferromagnétiques ou la magnétisation sont difficiles à effectuer, il est conseillé d'utiliser l'interaction entre le champ électromagnétique extérieur et le courant électrique conduit directement ou celui induit indirectement, ou bien d'utiliser l'interaction entre ces deux courants.

Remarques :

Si les fépoles sont des systèmes dans lesquels sont introduites des particules ferromagnétiques, alors les époles sont des systèmes dans lesquels au lieu des particules ferromagnétiques agissent (ou interagissent) les courants électriques.

L'évolution des époles, tout comme l'évolution des fépoles, suit le schéma commun : "époles simples – époles complexes – époles basés sur l'environnement – dynamisation – structuration – accord des rythmes".

Standard 2.4.12 : Liquide électrorhéologique

Une forme particulière des époles est un liquide électrorhéologique (une suspension d'une fine poudre de quartz dispersée, par exemple, dans le toluène) à la viscosité contrôlable. Si le liquide ferromagnétique ne peut pas être utilisé, alors on peut se servir du liquide électrorhéologique.

Classe 3 : Transition vers un super-système et au micro-niveau

Sous-classe 3.1 : Transition vers les bi- et poly-systèmes

Descriptif :

A côté du perfectionnement à l'intérieur du système (le cas des standards de la classe 2), il existe une voie d'évolution à l'extérieur du système. A toute étape d'évolution intérieure, le système peut être réuni avec d'autres systèmes et former un super-système possédant de nouvelles propriétés.

Standard 3.1.1 : Transition systémique 1-a : formation de bi- et poly-systèmes

On peut accroître l'efficacité d'un système (à toute étape de son évolution) par la transition systémique 1-a, notamment, en réunissant le système avec un autre système (ou systèmes) en bi- ou poly-système plus complexe.

Remarques :

1. Pour former un bi- ou poly-système, dans le cas le plus simple, on réunit deux substances S1 et S2 ou plus (vépoles avec bi- et poly-substances).

2. On peut considérer le standard 2.2.2 aussi comme la transition vers les poly-systèmes (mais plus précisément il faut le considérer comme une transition vers les systèmes plus poly-systémiques). L'unité des contraires : "division et union" aboutissent au même résultat, notamment à la formation de bi- et poly-systèmes. Il est possible que se forment des systèmes vépoles dans lesquels les champs et les substances se multiplient simultanément. Parfois se multiplie un couple (C - S) ou un vépole en général.

Standard 3.1.2 : Développement des liens dans les bi- et poly-systèmes

On parvient à accroître l'efficacité des bi- et poly-systèmes synthétisés, tout d'abord, en développant des liens entre les éléments dans ces systèmes.

Les bi- et poly-systèmes "nouveau-nés" ont souvent le lien "zéro", c'est à dire ils ne représentent qu'un "tas" d'éléments. L'évolution tend vers le renforcement des liens entre ses éléments.

D'autre part, les éléments dans les systèmes "nouveau-nés" sont parfois liés entre eux. En ce cas, l'évolution est dirigée vers l'augmentation du degré de la dynamisation des liens.

Standard 3.1.3 : Transition systémique 1-b : augmentation de la différence entre les éléments

L'efficacité des bi- et poly-systèmes accroît avec l'augmentation de la différence entre les éléments du système (transition systémique 1-b) : transition des mêmes éléments (lot de mêmes crayons) vers des éléments avec des caractéristiques décalées (lot de crayons de couleur), puis vers des éléments différents (boîte à compas) et enfin vers des combinaisons inverses du type "élément et anti-élément" (crayon avec la gomme).

Standard 3.1.4 : Réduction des bi- et poly-systèmes

L'efficacité des bi- et poly-systèmes s'accroît lors de la réduction des systèmes en commençant par l'élimination des parties auxiliaires. Les bi- et poly-systèmes réduits complètement redeviennent mono-systèmes. Le cycle peut se reproduire à un autre niveau.

Standard 3.1.5 : Transition systémique 1-c : propriétés contradictoires du tout et des parties

On peut accroître l'efficacité des bi- et poly-systèmes en répartissant les propriétés incompatibles entre le système et ses parties. C'est la transition systémique 1-c. On utilise un système constitué de deux niveaux où tout le système possède la propriété P et ses parties (particules) ont la propriété anti P.

Sous-classe 3.2 : Transition vers les super-systèmes

Descriptif :

Il existe deux voies de transition vers des systèmes principalement nouveaux : la transition vers un super-système ("vers le haut") – la sous-classe 3.1 et la transition vers l'utilisation des sous-systèmes de niveau hiérarchique inférieur ("vers le bas") – la sous-classe 3.2.

Standard 3.2.1 : Transition systémique 2 : transition au micro-niveau

On peut accroître l'efficacité d'un système (à toute étape de son évolution) par la transition systémique 2 : du macro-niveau au micro-niveau, c'est-à-dire remplacer le système ou une de ses parties par une substance qui, en interaction avec un champ, est capable d'accomplir l'action exigée.

Remarques :

- 1. La transition vers le micro-niveau est possible à n'importe quelle étape d'évolution du système.*
- 2. La transition "macro-micro" est une notion généralisée. Il existe plusieurs "micro-niveaux" (cellule, molécules, atomes, etc.). En résumé, il existe différents types de transition au*

micro-niveau, et également, plusieurs transitions d'un micro-niveau à un autre micro-niveau inférieur.

Classe 4 : Standards de détection et de mesure des systèmes

Sous-classe 4.1 : Détours

Descriptif :

Les mesures et les détections dans les systèmes servent l'action essentielle "de mesure". C'est pourquoi il est souhaitable de modifier l'action principale de telle sorte qu'elle exclue la nécessité (ou qu'elle la réduise au minimum) de l'action de mesure et de détection, bien entendu, non au détriment de la précision.

Standard 4.1.1 : Modification des systèmes au lieu de détection ou de mesure

Si le problème exige une détection ou une mesure, alors il est utile de modifier le système à la manière qu'il ne soit plus de résoudre ce problème.

Standard 4.1.2 : Utilisation de copies

Si l'on a un problème de mesure ou de détection et si on ne peut pas appliquer le standard 4.1.1, alors il est utile de remplacer les opérations directes avec l'objet par des opérations avec sa copie ou sa photo.

S'il faut comparer l'objet à l'étalon pour mettre en évidence les différences, alors on résout le problème en superposant les images de l'objet et de l'étalon, sachant que la couleur de l'image optique de l'objet doit être contraire à celle de l'étalon ou de son image. Par analogie, on résout les problèmes de mesure avec un étalon ou son image.

Standard 4.1.3 : Mesure comme deux détections successives

Si l'on a un problème de mesure et si l'on ne peut pas appliquer les standards 4.1.1 et 4.1.2, alors il est utile de le transformer en un problème de détection successive des modifications.

Remarques :

On mesure toujours avec un certain degré de précision. C'est pourquoi dans les problèmes de mesure, même s'il s'agit d'une mesure continue, on peut toujours distinguer un acte élémentaire de mesure, constitué de deux détections successives.

Le passage de la notion vague "mesure" au modèle clair "deux détections successives" simplifie nettement le problème.

Sous-classe 4.2 : Synthèse des systèmes de mesure

Descriptif :

Dans la synthèse des systèmes de mesure on voit apparaître une tactique typique à la synthèse des systèmes "de modification". Elle consiste à achever le vépole par tous les moyens en introduisant les substances et/ou les champs qui manquent. La synthèse des vépoles de mesure se distingue par le fait que la structure du vépole doit assurer l'obtention d'un champ à la sortie.

Standard 4.2.1 : Vépole "de mesure"

Si le système non-vépole résiste à la détection ou à la mesure, alors on résout le problème en construisant le vépole simple ou double avec un champ à la sortie.

Standard 4.2.2 : Vépole complexe "de mesure"

Si le système (ou sa partie) résiste à la détection ou à la mesure, alors on résout le problème par la transition vers le vépole complexe intérieur ou extérieur en introduisant des additifs facilement détectables.

Standard 4.2.3 : Vépole "de mesure" basé sur l'environnement

S'il est difficile de détecter ou de mesurer le système à un moment donné et s'il n'y a pas de possibilité d'introduire dans l'objet des additifs qui créent un champ facilement détectable et mesurable, alors on doit les introduire dans l'environnement. En fonction du changement de son état, on peut juger du changement de l'état de l'objet.

Standard 4.2.4 : Obtention des additifs dans l'environnement

Si l'on ne peut pas introduire de nouveaux additifs dans l'environnement, selon le standard 4.2.3, alors on peut obtenir ces additifs dans l'environnement lui-même, notamment en le décomposant ou en changeant sa phase .

Le rôle de ces additifs peut être exécuté par les bulles de gaz ou de vapeur obtenues par électrolyse, cavitation, etc.

Sous-classe 4.3 : Renforcement des vépoles "de mesure"

Descriptif :

On peut renforcer les vépoles de mesure par les effets physiques et en accordant leurs rythmes.

Standard 4.3.1 : Utilisation des effets physiques

On peut accroître l'efficacité de la détection et de la mesure d'un système vépole en utilisant les effets physiques.

Particulièrement, il est souhaitable que les substances forment dans le vépole un thermocouple qui signale "gratuitement" l'état du système. "Le champ émetteur" peut être également obtenu par induction.

Standard 4.3.2 : Utilisation de la résonance de l'objet contrôlé

S'il est impossible de détecter ou de mesurer directement un changement qui a lieu dans le système et de faire passer un champ à travers le système, alors on résout le problème en provoquant dans le système (dans tout le système ou dans une de ses parties) les vibrations de résonance. En fonction du changement de leurs fréquences, on peut définir les modifications qui ont lieu dans le système.

Standard 4.3.3 : Utilisation de la résonance d'un objet conjoint

S'il est impossible d'appliquer le standard 4.3.2, on juge de l'état d'un système selon le changement de la fréquence propre à l'objet (l'environnement) relié au système contrôlé.

Sous-classe 4.4 : Transition vers les systèmes fépoles

Descriptif :

Les vépoles de mesure ont une tendance particulièrement prononcée à passer dans la suite fépole.

Standard 4.4.1 : "Proto Fépoles de mesure"

Les vépoles contenant des champs non-magnétiques ont une tendance à passer aux "protofépoles", c'est-à-dire aux vépoles comprenant une substance et un champ magnétique.

Standard 4.4.2 : Fépole "de mesure"

S'il faut augmenter le degré d'efficacité de la détection ou de la mesure au moyen des systèmes "protofépoles" ou vépoles, alors il faut passer aux fépoles en remplaçant une des substances par des particules ferromagnétiques (ou en les ajoutant) et en détectant ou en mesurant le champ magnétique.

Standard 4.4.3 : Fépole complexe "de mesure"

S'il faut accroître l'efficacité de la détection ou de la mesure d'un système par la transition vers le fépole et si le remplacement de la substance par les particules ferromagnétiques est inadmissible, alors on effectue la transition vers le fépole par la construction du fépole complexe en introduisant des additifs dans la substance.

Standard 4.4.4 : Fépole "de mesure" basé sur l'environnement

S'il faut accroître l'efficacité de la détection ou de la mesure d'un système au moyen de la transition du vépole au fépole et si l'introduction des particules ferromagnétiques est inadmissible, alors il est conseillé de les introduire dans l'environnement.

Standard 4.4.5 : Utilisation des effets physiques

S'il faut accroître l'efficacité d'un système vépole de mesure, il est nécessaire d'utiliser les effets physiques. Par exemple, pour le passage par le point de Curie, on peut utiliser les effets d'Hopkinson et de Barkhausen, l'effet magnétoélastique, etc.

Sous-classe 4.5 : Voie d'évolution des systèmes de mesure

Descriptif :

L'évolution des vépoles de mesure s'effectue par les transitions systémiques ordinaires, cependant elle se caractérise par des particularités spécifiques.

Standard 4.5.1 : Transition vers les bi- et poly-systèmes

On peut accroître l'efficacité d'un système de mesure à toute étape de son évolution par la transition vers un bi- ou poly-système.

Standard 4.5.2 : Voie d'évolution

L'évolution des systèmes de mesure emprunte la voie suivante : "mesure de la fonction – mesure de la première dérivée – mesure de la deuxième dérivée".

Classe 5 : Standards d'application des standards

Sous-classe 5.1 : Introduction de substances

Descriptif :

Pour construire, transformer et détruire les vépoles, on est souvent amené à introduire de nouvelles substances. De leur introduction il en résulte des difficultés techniques, car elle diminue le degré d'idéalité du système. C'est pourquoi il faut introduire des substances "sans les avoir introduites réellement" et utiliser divers détours.

Standard 5.1.1 : Détours

S'il est nécessaire d'introduire dans un système une substance et si c'est interdit, selon les conditions du problème, ou impossible, selon les conditions de travail du système, il faut chercher des détours :

- on utilise le "vide" au lieu de substance ;
- on utilise un additif extérieur au lieu d'un additif intérieur ;
- on utilise un champ au lieu de substance ;
- on introduit un additif particulièrement actif en petites doses ;
- on introduit un additif ordinaire en très petites doses, mais on le concentre dans certaines parties de l'objet ;
- on introduit un additif provisoirement ;
- on utilise au lieu de l'objet sa copie (le modèle) qui permet l'introduction d'additifs ;
- on introduit un additif sous forme d'une combinaison chimique d'où on l'élimine par la suite ;
- on obtient un additif en décomposant l'environnement ou l'objet lui-même par électrolyse ou bien en changeant la phase d'une partie de l'objet ou de l'environnement.

Standard 5.1.2 : "Dédoublage" de la substance

Si un système résiste aux changements exigés et si les conditions du problème ne permettent pas de remplacer l'instrument ou d'introduire des additifs, alors au lieu de l'instrument on utilise l'objet en le divisant en parties qui interagissent entre elles.

Si un flux de particules concassées jusqu'aux dimensions minimales fait partie du système et s'il faut augmenter le degré de contrôle de ces particules, alors il est conseillé de diviser le flux en parties de même polarité ou de polarité opposée. Si tout le flux est polarisé de même signe, une de ses parties doit porter le signe opposé.

Standard 5.1.3 : Auto-élimination des substances usagées

La substance introduite dans le système (après avoir été utilisée) doit disparaître ou devenir indiscernable de la substance qui se trouvait auparavant dans le système ou dans l'environnement.

Standard 5.1.4 : Introduction de grandes quantités de substances

S'il faut introduire une grande quantité de substance et si c'est interdit selon les conditions du problème, ou inadmissible selon les conditions de travail du système, alors on utilise en tant que substance le vide sous forme de constructions gonflables ou de mousse.

Remarques :

L'utilisation des constructions gonflables est un standard au macro-niveau. L'utilisation de la mousse est le même standard au micro-niveau.

On utilise souvent le standard 5.1.4 avec d'autres standards.

Sous-classe 5.2 : Introduction des champs

Descriptif :

Pour construire, transformer et détruire les vépoles, il est souvent nécessaire d'introduire de nouveaux champs. Pour ne pas compliquer le système, il faut utiliser les standards de la sous-classe 5.2.

Standard 5.2.1 : Deuxième utilisation des champs

S'il faut introduire un champ dans un système vépole, il faut, tout d'abord, utiliser tous les champs qui s'y trouvent déjà, dont les porteurs sont les substances contenues dans le système.

Standard 5.2.2 : Introduction des champs de l'environnement

S'il faut introduire un champ et s'il est impossible d'utiliser le standard 5.2.1, alors il faut utiliser les champs contenus dans l'environnement.

Standard 5.2.3 : Utilisation des substances susceptibles de devenir les sources de champs

S'il faut introduire un champ dans un système et s'il est impossible d'utiliser les standards 5.2.1 et 5.2.2, alors il faut utiliser les champs qui peuvent être générés ou portés par les substances contenues déjà dans le système ou dans l'environnement.

Si le système contient des substances ferromagnétiques utilisées uniquement d'une manière mécanique, il faut également se servir de leurs propriétés magnétiques pour obtenir des effets supplémentaires, notamment, améliorer l'interaction entre les éléments, obtenir l'information sur le travail et l'état du système, etc..

Sous-classe 5.3 : Transition de phase

Descriptif :

Des exigences contradictoires aux substances et aux champs introduits peuvent être satisfaites par l'utilisation de transitions de phase.

Standard 5.3.1 : Transition de phase 1 : changement de la phase

L'efficacité d'utilisation des substances sans en introduire d'autres peut être augmentée par la transition de phase du premier degré, c'est-à-dire par le changement de phase de la substance donnée.

Standard 5.3.2 : Transition de phase 2 : la phase duale

Les propriétés "duales" peuvent être obtenues par la transition de phase du deuxième degré, c'est-à-dire par l'utilisation de substances capables de passer d'un état à un autre en fonction des conditions de travail.

Standard 5.3.3 : Transition de phase 3 : utilisation des phénomènes concomitants

On peut accroître l'efficacité d'un système par la transition de phase 3, c'est-à-dire par l'utilisation des phénomènes accompagnant la transition de phase.

Standard 5.3.4 : Transition de phase 4 : transition vers l'état biphasé

Les propriétés "duales" d'un système peuvent être obtenues par la transition de phase 4, c'est-à-dire par la substitution d'un état monophasé par un état biphasé.

Standard 5.3.5 : Interaction des phases

On peut accroître l'efficacité des ST qui résultent de la transition de phase 4 par l'interaction (physique et chimique) entre les parties ou les phases du système.

Sous-classe 5.4 : Particularités d'utilisation des effets physiques

Descriptif :

De nombreux standards prévoient l'utilisation des effets physiques ou peuvent être utilisés en combinaison avec eux. Dans ce cas, il est nécessaire de tenir compte de certains procédés qui augmentent l'efficacité d'utilisation des effets physiques.

Standard 5.4.1 : Transitions auto-contrôlables

Si l'objet doit se trouver périodiquement dans différents états physiques, alors il faut effectuer la transition à l'aide de l'objet lui-même, en utilisant des transformations physiques réversibles. On peut utiliser, par exemple, les transitions de phase suivantes : ionisation – recombinaison, dissociation – association, etc..

Standard 5.4.2 : Renforcement du champ à la sortie

S'il faut obtenir une action forte à la sortie en ayant une action faible à l'entrée, il est nécessaire de mener la substance transformatrice à l'état proche de l'état critique. Ainsi, l'énergie s'accumule dans la substance et le signal d'entrée joue le rôle de la "gâchette".

Sous-classe 5.5 : Standards expérimentaux

Standard 5.5.1 : Obtention des particules de substance par décomposition

Si, pour résoudre le problème, on a besoin de particules de la substance (par exemple, des ions) et si leur obtention directe est impossible selon les conditions du problème, alors il faut obtenir les particules exigées en décomposant une substance d'un niveau structurel supérieur (par exemple, les molécules).

Standard 5.5.2 : Obtention des particules de substance par combinaison

Si, pour résoudre le problème, on a besoin de particules de la substance (par exemple, des molécules) et s'il est impossible de les obtenir directement ou selon le standard 5.5.1, alors il faut obtenir les particules exigées en achevant leur construction ou en combinant les particules d'un niveau structurel inférieur (par exemple, les ions).

Standard 5.5.3 : Application des standards 5.5.1 et 5.5.2

Quand on applique le standard 5.5.1, la voie la plus simple est la destruction du niveau "entier" ou "excédentaire" (ions négatifs) juste supérieur. Quand on applique le standard 5.5.2, la voie la plus simple est la recombinaison du niveau "non-entier" juste inférieur.

ANNEXE-II. ARIZ-71 EXTRAITS, SELON [ALTSHULLER, 1973]

Partie 1. « Choix du problème »

Etape 1.1. Détermination de l'objectif final de la solution

- a. Quel est l'objectif technique (quelle caractéristique de l'objet doit être changée) ?
- b. Quelle caractéristique de l'objet ne peut pas être changée durant le processus de résolution de problème ?
- c. Quel est l'objectif économique de la solution (quelle dépense sera réduite si le problème est résolu) ?
- d. Quelles sont les dépenses acceptables ?
- e. Quelle est la caractéristique technique / économique principale à améliorer ?

Etape 1.2. Recherche d'une approche de contournement. Imaginez que le problème ne peut être résolu. Quel autre problème, plus général, peut être résolu afin d'atteindre le résultat requis.

Etape 1.3. Déterminer quel problème, l'original ou celui de contournement, est le plus pertinent à résoudre

- a. Comparez le problème original avec les tendances (ou directions d'évolution) de l'industrie considérée.
- b. Comparez le problème original avec les tendances (ou directions d'évolution) de l'industrie leader du marché.
- c. Comparez le problème de contournement avec les tendances (ou directions d'évolution) de l'industrie considérée.
- d. Comparez le problème de contournement avec les tendances (ou directions d'évolution) de l'industrie leader du marché.
- e. Comparez le problème original avec celui de contournement. Choisir celui à résoudre.

Etape 1.4. Déterminer les caractéristiques requises, de manière quantitative.

Etape 1.5. Introduire les corrections temporelles aux caractéristiques quantitatives.

Etape 1.6. Déterminer les requis pour les conditions spécifiques dans lesquelles l'invention sera amenée à fonctionner.

- a. Considérer les conditions spécifiques de production du produit : en particulier, le degré de complexité acceptable.
- b. Considérer l'échelle des futures applications.

Partie 2. « Définition plus précise du problème »

Etape 2.1. Définir le problème plus précisément en utilisant les informations issues des brevets.

- a. De quelle manière le problème se rapproche de ceux résolus dans les brevets ?
- b. De quelle manière les problème similaires sont-ils résolus dans les industries leader du marché ?
- c. De quelle manière les problème opposés sont-ils résolus ?

Etape 2.2. Utiliser l'opérateur « Dimension Temps Coût ».

- a. Imaginer de changer les dimensions d'un objet de sa valeur originale à 0. Le problème peut-il être résolu ? Si oui, comment ?
- b. Imaginer de changer les dimensions d'un objet de sa valeur originale à l'infini. Le problème peut-il être résolu ? Si oui, comment ?
- c. Imaginer de changer la durée d'un processus de sa valeur originale à 0. Le problème peut-il être résolu ? Si oui, comment ?
- d. Imaginer de changer la durée d'un processus de sa valeur originale à l'infini. Le problème peut-il être résolu ? Si oui, comment ?
- e. Imaginer de changer le coût d'un objet ou d'un processus de sa valeur originale à 0. Le problème peut-il être résolu ? Si oui, comment ?
- f. Imaginer de changer le coût d'un objet ou d'un processus de sa valeur originale à l'infini. Le problème peut-il être résolu ? Si oui, comment ?

Etape 2.3. Décrire les conditions du problème (sans utiliser de termes spécifiques, et sans décrire ce qui doit être fait ou développer) en deux phrases selon le patron suivant :

- a. « Soit un système consistant en (décrire les éléments). »
- b. « L'élément (décrire l'élément), dans les conditions (décrire les conditions), produit l'effet indésirable (décrire l'effet). »

Etape 2.4. Entrer les éléments de l'étape 2.3. dans un tableau.

Types d'éléments	Eléments
Eléments pouvant être changés, re-conçus ou retirés.	
Eléments difficiles à changer (sous les conditions du problème).	

Etape 2.5. Choisir parmi les éléments de l'étape 2.4., et parmi ceux pouvant être changés, l'élément le plus facile à changer, re-concevoir ou retirer.

- a. Si tous les éléments se valent, commencer par considérer un élément immobile (ils sont en général plus facile à modifier).
- b. S'il existe un élément, à l'étape 2.4., qui est connecté à l'effet indésirable, le choisir en dernier lieu.
- c. Si le système ne comporte que des éléments ne pouvant être changé, considérer un élément issu de l'environnement

Partie 3. « Phase analytique »

Etape 3.1. Formuler le Résultat Idéal Final.

- a. Sélectionner l'élément identifié au 2.5.
- b. Définir son action.
- c. Définir Comment ? il réalise cette action (en répondant à cette question, utiliser toujours le terme « de lui-même »).
- d. Définir Quand ? il réalise cette action.

e. Définir sous quelles conditions (limitations, requis, etc.) il réalise cette action.

Etape 3.2. Tracer deux figures : la situation initial (les conditions avant le RIF), et la situation idéale (les conditions une fois le RIF atteint).

Etape 3.3. Dans la figure idéale, identifier l'élément du 3.1.a. et faire ressortir la partie qui ne peut réaliser la fonction requise sous les conditions du problème.

Etape 3.4. Pourquoi cet élément ne peut-il pas (de lui-même) réaliser l'action requise ?

Etape 3.5. Sous quelles conditions cette partie peut-elle réaliser l'action requise ? (Quels paramètres cette partie doit-elle posséder ?)

Etape 3.6. Que faut-il faire pour que cet élément atteigne les caractéristiques décrites au 3.5. ?

Etape 3.7. Formuler le concept réalisable en pratique. Si il en existe plusieurs, les trier par ordre d'intérêt.

Etape 3.8. Produire un schéma de réalisation du concept.

ANNEXE-III. ARIZ-77, SELON [ARCISZEWSKI, 1988]

Partie 1. « Formulation du problème »

Etape 1.1. Identifier l'objectif ultime de la résolution de problème

Etape 1.2. Déterminer si cet objectif peut être atteint indirectement. Imaginez que le problème ne peut être résolu. Quel autre problème, plus général, peut être résolu afin d'atteindre le résultat requis.

Etape 1.3. Déterminer quel problème, l'original ou celui de contournement, est le plus pertinent à résoudre.

Etape 1.4. Déterminer les caractéristiques requises de description du système, de manière quantitative.

Etape 1.5. Modifier ces paramètres quantitatifs, en considérant les changements futurs et toute limitation de temps sur le processus de développement.

Etape 1.6. Déterminer les requis liés à la production.

Etape 1.7. Déterminer si le problème peut être résolu en recourant aux solutions standards.

Etape 1.8. Formuler le problème plus précisément en utilisant les informations issues des brevets.

Etape 1.9. Utiliser l'opérateur « Dimension Temps Coût ».

Partie 2. « Construction du modèle de problème »

Etape 2.1. Formuler le problème selon ses propres termes.

Etape 2.2. Identifier les deux composants conflictuels du système. S'il n'en existe qu'un, recourir aux transformations vépole.

Etape 2.3. Identifier les interactions entre les composants et les caractériser selon qu'elles sont utiles ou néfastes.

Etape 2.4. Ecrire le problème formulé, en indiquant les composants conflictuels et la contradiction technique.

Partie 3. « Phase analytique »

Etape 3.1. De tous les composants du système, sélectionner celui qui peut être changé, modifié ou remplacé aisément

Etape 3.2. Formuler le Résultat Idéal Final.

Etape 3.3. Identifier la partie du composant identifié au 3.1. qui ne peut satisfaire les requis fonctionnels imposés.

Etape 3.4. Ecrire la contradiction physique formulée.

ANNEXE-IV. ALGORITHM OF INVENTIVE PROBLEM SOLVING : ARIZ-85C

© G.S. Altshuller, 1956-1985

ATTENTION!

ARIZ is a complicated tool. Do not apply it to solve new practical problems without at least for 80 academic hours of preliminary study.

ARIZ is a tool for thinking, but not instead of thinking. Do not hurry! Consider each step carefully. Besides, note obligatory all considerations (at margins), appearing in process of the problem's solving.

ARIZ is a tool for solving non-typical problems. Let's check: may your problem be solved using the System of Standard Solutions for Inventive Problems (Inventive Standards)?

PART 1. ANALYZING THE PROBLEM

1.1. formulate the mini-problem

Formulate the "mini-problem" conditions (without special terms) accordingly to the following pattern:

A technical system for <state the purpose of the system> includes <list the main parts of the system>.

Technical contradiction 1 (TC-1): (identify).

Technical contradiction 2 (TC-2): (identify).

It is necessary with minimum changes to the system, to <state the required result>.

Example:

The technical system <for receiving of radio waves> includes <radio telescope's antenna, radio waves, lightning, and lightning rods>.

TC-1: if there are many lightning rods, then they reliably protect the antenna from lightning, but absorb radio waves.

TC-2: if there are few lightning rods, then there is no remarkable absorption of radio waves, but the antenna is not protected from lightning.

It is necessary with minimum changes to the system, to <protect the antenna from lightning without radio waves' absorption>.

(For this definition it should be replaced the special term "lightning rod" to "conducting rod", "conducting column" or simply "conductor").

Comments:

1. The mini-problem is obtained from the initial problem situation by introducing the restrictions: "Everything in the system remains unchanged or becomes simpler, while the required action (or property) appears or a harmful action (or property) disappears. The transition from the problem situation to the mini-problem does not mean intention to solve small problem. Quite the contrary, introduction of additional requirements (result has to be achieved "without nothing") directs to intensify conflict, and cuts the paths to compromise solutions.

2. While formulating step 1.1, it should be indicated not only the technical parts of the system but also the natural ones that interact with the system. In the problem of protection the telescope antenna the natural parts are lightning and received radio waves (if they are emitted by natural cosmic objects).

3. **The Technical contradictions (TC)** denote the interaction in the technical system when the useful actions create the harmful ones. In other words, introduction (or improvement) a useful action or elimination (or reducing) a harmful action cause degradation (in particular, complication) either all or part of the system as result. Technical contradictions are formulated by identifying (write down) one state of a system element with explanation of both the good and bad results of this state. Then the opposite state of the system element is identified, along with its associated explanation. Sometimes for problem situation is given only the *product*. There is no a technical system (a *tool*) and therefore there is no clear technical contradiction (TC). In this case the TC can be identified by considering two states of the *product*, even if one of the states is impossible to achieve.

For example a problem is given: "How to watch the micro-particles in a sample of an optically clean liquid with the naked eye; the particles are so small, that the light flows around them?"

TC-1: if the particles are small, then the liquid is optically clean, but it is impossible to watch the particles by the naked eye.

TC-2: if the particles are big, then they are well watched, but the liquid is not optically clean, this is unacceptable consequence.

It seems that the problem definition excludes considering TC-2: it is prohibited to change product! In fact we will consider only TC-1 for this problem, but TC-2 will give us an additional requirements for product: small particles have to be small and have to be also become large.

4. To reduce mental inertia, special terms associated with the *tool* and environment should be replaced with easy words, because special terms:

- impose old concepts about working principles of the tool: for example, "the icebreaker breaks ice", when it possible to move through ice without breaking its;
- can hide certain properties of the elements described in the problem situation: "mould" (to form concrete) this is not plain "wall", but "iron wall";
- narrow the range of possible states of a substance: the term "paint" impose to imaging liquid or solid paint, although paint may be gaseous.

1.2. define the conflicting elements

Identify and write down the conflicting pair: a *product* and a *tool*.

Rule 1.

If the tool according problem situation conditions can be in two states, it is necessary to indicate the both of these states.

Rule 2.

If the problem situation include several similar pairs of interacting elements, it is enough to consider only one pair.

Example:

Product: *lightning and radio waves.*

Tool: *conducting rods (many rods or a few rods).*

Comments:

5. **The product** is the element that need to be processed (manufactured, moved, changed, improved, protected from a harmful influence, revealed measured etc.) according the problem conditions. For problems about detection and measurement some element considered as *tool* (according its base function), can be considered as *product* (e.g., grinding wheel).

6. **The tool** is the element that directly interacts with the *product* (e.g., mill rather than a milling machine; fire rather than a burner). In particular, a part of the environment can be considered as a *tool*. The standard parts from which the product is assembled can be considered as a *tool* too (e.g., meccano this is tool to create of various "product").

7. One of the elements in the conflicting pair can be doubled. For instance, two different *tools* are given, and they have to act on the *product* simultaneously, where one *tool* interferes with the other. Or two *products* are given, and they have to be processed with the same *tool*, where one *product* interferes with another.

1.3. describe graphic models for technical contradictions

Build graphic models for conflicts for TC-1 and TC-2 using Table 1. Typical Graphic Models of Technical Contradictions.

Example:

TC-1: Many conducting rods

TC-2: A few conducting rods

Comments:

8. The Table 1 contains graphic models of typical conflicts. It is acceptable to use original (atypical) graphic models if they describe the meaning of conflict clearly.

9. Some problems can be described by multi-linked graphic models, for example:

Such models can be converted as two one-linked graphic models:

If B is considered as a *modified product* or the basic property (or state) of A is transferred on B.

10. The conflict can be considered *in time* as well as *in space*.

11. Steps 1.2 and 1.3 specify the general problem situation description. Therefore it is necessary to return to the step 1.1 after step 1.3 and to check if there is any discordance in the sequence 1.1 - 1.2 - 1.3. If a discordance exists, it is necessary to remove its and set up the sequence.

1.4. select a graphic model for further analysis

From the two graphic models of conflict it is necessary to choose the one, which provides the best performance for the Main Manufacturing Process (i.e., the main function of technical system as indicated in the problem description). Indicate what the Main Manufacturing Process (MMP) is.

Example:

In the problem of protection the radio telescope antenna the Main Manufacturing Process is receiving of radio waves. Therefore it should choose the TC-2: in this case the conducting rods do not absorb the radio waves.

Comments:

12. When choosing one of the two graphic conflict model, it is chosen one state of the *tool* from its two opposite states. Further problem solving efforts should be related to this state. It is prohibited to replace *few conductors* by some *optimal number of conductors*. *ARIZ requires intensifying the conflict rather than smoothing it over*. Holding one state of the tool, it is necessary in future to achieve the required positive property for the tool in this state (needed property inherent in another state of the tool). *There are few conductors and we will not increase*

their number, but as a result of solving problem the lightning has to be eliminated in the same manner as if there were many conductors.

13. From time to time, it is difficult to define MMP for measurement problems. Ultimately, measurements are almost always performed for modification purpose, i.e. to process a product, to produce something, etc. Therefore the MMP for measurement problems is the MMP of the whole system, not just its measuring part.

For instance, it is necessary to measure gas pressure inside electric bulbs. MMP – to produce electric bulbs rather than to measure gas pressure.

However, some problems of measurement for scientific purposes can be considered as exception.

1.5. intensify the conflict

Intensify the conflict by indicating the extreme state (action) of the elements.

Rule 3.

Most problems contain the following types of conflicts: "many elements" versus "few elements" ("strong element" versus "weak element" etc.). The conflict about "few elements", should be converted into form "no elements" ("absent element") only.

Example:

Let's consider that instead of "few conductors" there is "absent conductor" in TC-2.

1.6. describe the problem model

Formulate the Problem Model to indicate the following:

- 1) the conflicting pair;
- 2) the intensified definition of conflict;

3) what has to do the X-element introduced to solve problem (what the X-element should keep, eliminate, improve, provide, etc.).

Example:

It is given an absent conductor and a lightning. The absent conductor does not produce interference (when antenna receives the radio waves) and does not provide protection from lightning.

It is required to find an X-element that will keep ability of the absent conductor does not produce interference (for antenna) and will provide protection from lightning.

Comments:

14. The Problem Model is a type of abstraction with artificially selected only some of the elements of the technical system. Other elements are implied only. *For instance, in the Problem Model of antenna protection it was selected only two of the four elements (the antenna, radio waves, the conductor and a lightning). The another two elements are just implied - they could be ignored at all.*

15. After step 1.6 it is required to return on step 1.1 and check the creation logic of the Problem Model. Generally, it is possible to define more exactly the chosen graphic model of conflict by indicating the action of the X-element.

16. The X-element is not necessarily the new material part of the system. The X-element this is some modification in system, something basically unknown - X. For instance, it may be temperature changes or phase state changes for some part of system or the environment.

1.7. apply the inventive standards

Check possibility to apply the System of Standard Solutions for Inventive Problems for solving the Problem Model. If the problem has not been solved, go to Part 2 of ARIZ (i.e., steps 2.1 - 2.3). If the problem has been solved, go to Part 7 of ARIZ. Although in this case, it is recommended to continue the analysis through Part 2 too.

Comment:

17. The analysis performed in Part 1 of ARIZ and developing of the Problem Model makes problem clear and in many cases allows identifying standard (typical) properties for non-typical problems. It gives a possibility to use the Inventive Standards more effectively than for initial problem situation description.

PART 2. ANALYZING THE PROBLEM MODEL

The main purpose of Part 2 is to identify available resources (space, time, substances, and fields) that may be useful for problem solving.

2.1. define the operational zone (oz)

Analyze and describe the Operational Zone (OZ).

Comment:

18. In the simplest case the Operational Zone this is the space where the conflict indicated in Problem Model appears.

Example:

In the problem about protection of antenna the OZ is the space previously occupied by the lightning conductor, i.e., mentally defined as the "empty" rod, or "empty" column.

2.2. define the operational time (ot)

Analyze and describe the Operational Time (OT).

Comment:

19. The operational time there are available resources of time: the time when conflict occurs - T1 and the time before the conflict - T2. A conflict (especially high-velocity, short-term) can usually be eliminated (prevented) during T2.

Example:

In the problem about protection of antenna the Operational Time consists of the time T1' (during lightning discharge) and the time T1'' (before next lightning discharge). The T2 is not considered here.

2.3. define the substance-field resources (sfr)

Define the Substances and Fields Resources (SFR) of the analyzing system, the environment, and the product. Compose a list of SFR.

Comments:

20. Substance and Field Resources (SFR) there are substances and fields that already exist or may be easily obtained according to the problem conditions. There are three types of SFR:

1. System (internal) resources:

- a) SFR of the tool;
- b) SFR of the product.

2. Available (external) resources:

- a) SFR of the environment that are particular for the problem conditions, for instance, in the problem about small particles in an optically clean liquid, water is SFR;
- b) SFR common for any environment, including "background" fields, for instance gravity and magnetic field of the Earth.

3. SFR of super-system:

- a) waste materials of some outside system (if available according problem conditions);
- b) very cheap outside resource, which cost may be ignored.

During solving a Mini-Problem it is necessary to achieve needed result with minimum expenditure of SFR. Therefore, the utilization of internal (system) SFR should first be considered. However, when it is necessary to develop solution concepts and/or solve problem about forecasting (i.e., Maxi-Problems), the maximum of various SFR should be considered.

21. It is known that the product is the unchangeable element - so what kinds of resources can be it has? Indeed the product can not be changed, i.e. it is unsuitable to change product while solving the Mini-Problem. However, sometimes the product can:

- a) change itself;

- b) allow consumption (i.e. modification) of part, when the product amount is unlimited (for instance, wind etc.);
- c) allow transition to the super-system (for instance, brick has not changed, but house has changed);
- d) allow application of micro-level structures;
- e) allow being combined with "nothing", i.e. with void;
- f) allow temporary modification.

Therefore, the product can be considered as SFR only in rare cases if product can be easily modified "without modification".

22. The SFR are available resources and thus they should be utilized first. If there are not enough available resources other substances and fields can be considered. The analysis of SFR in step 2.3 is preliminary.

Example:

In the problem about protection of antenna the "absent lightning conductor" is considered. Therefore the SFR contain only substances and fields of environment. In this case SFR this is the air.

PART 3. DEFINING IDEAL FINAL RESULT (IFR) AND PHYSICAL CONTRADICTION (PhC)

As result of applying Part 3 should be formulated the image of Ideal Final Result (IFR). The Physical Contradiction (PhC) that prevents from achievement of IFR should be identified too. The ideal solution is not always achievable, but the IFR indicates the direction to the most powerful solution.

3.1. formulate ifr-1

Formulate and describe the IFR-1 using the following pattern:

The X-element, without complication of a system and without harmful side effects, eliminates <to indicate the harmful action> during the <Operational Time> inside the <Operational Zone>, and keeping the tool's ability to provide <to indicate the useful action>.

Example:

The X-element without complication of a system and without harmful side effects, eliminates

<"non-attraction" of lightning by the absent conducting rod> during the <Operational Time>

inside the <Operational Zone>, and keeping the tool's ability to provide <not hinder the antenna's reception>.

Comment:

23. There are other conflicts besides the conflict "a harmful action is associated with a useful one", for instance "introducing a new useful action results in complicating the system" or "one useful action is incompatible with another one". Therefore the formulation of IFR, identified in step 3.1 this is just a pattern to write down the IFR.

Basic meaning for any definition of the IFR is the following: a useful feature's obtaining (or a harmful feature's eliminating) should be achieved without other features' deterioration (or a harmful feature's appearing).

3.2. intensify definition of ifr-1

Intensify the formulation of IFR-1 by introducing additional requirement:

the introduction of new substances and fields into system is prohibited, it is necessary to use the SFR only.

Example:

There is no tool in the Problem Model about the antenna protection ("absent lightning conductor"). According to Comment 24 it should be introduced the environment into definition of IFR-1, i.e. it is necessary to replace an X-element by the word "air" (or, more exactly: "the air column where the absent lightning conductor was").

Comment:

24. According to Comments 20 and 21, SFR should be considered in the following order during problem solving process:

- SFR of the tool (system/internal resources);
- SFR of the environment (available/external) resources);
- SFR of super-system;
- SFR of product (if it is not prohibited by Comment 21).

The above types of resources (SFR) determine four directions of further analysis.

In practice, the problem conditions cut off some directions. When solving the Mini-Problem, it is enough to develop analysis up to the point where a solution concept is obtained; for instance, if a concept was obtained on the "tool line" (first line, above), then the other directions need not be considered. When solving the Maxi-Problem, it is necessary to consider all available directions. Thus, if you have found a concept on the "tool line", it should be considered the directions for environment SFR, for super-system SFR, and for product SFR.

When mastering ARIZ, a sequential, linear analysis is gradually being replaced by parallel analysis: ability to transfer the solution concept from one "line" to another. This kind of ability called a "multi-screen thinking": possibility to analyze changes for super-system, system, and subsystems simultaneously.

ATTENTION!

The problem solving is accompanied by breaking down of old conceptions. New concepts appear, which often can not be adequately described in words. For instance, how to describe a property of paint to dissolve, without dissolving (to paint, without painting)?..

Applying ARIZ, it is necessary to describe analysis using simple, non-technical, even "childish" words, avoiding special terms (they increase the mental inertia).

3.3. identify the physical contradiction for the macro-level

Identify and describe the Physical Contradiction at **macro-level** using the following pattern:

the <Operational zone>, during the <Operational time>, has to... <indicate physical macro-state, for example "hot"> in order to perform <indicate one of the conflicting actions> and does not have to <indicate the opposite physical macro-state, for example "cold"> to perform <to indicate another conflicting action or requirement>.

Comments:

25. The **Physical contradiction** this is opposite requirements to physical state of Operational Zone.

26. If it is difficult to describe a complete definition for the Physical Contradiction, it is allowed to compose the *brief* PhC according to the following pattern:

The element (or a part of it in the operational zone) has to be <feature> to perform <indicate> and does not have to be <feature> to perform <indicate>.

Example:

the <air column>, during the Operational Time has to be <conducting> to <remove the lightning> and does not have to be <conducting> to <prevent absorption of the radio waves>.

This definition suggests the answer: the air column has to be electrically conductive when a lightning discharge and does not have to be conductive when all the rest of the time. Lightning

discharge occurs relatively seldom and it acts quickly. The Law of coordination the rhythm of systems parts: periodicity of the lightning conductor appearance has to be same as the periodicity of lightning strike.

Obviously, this is not a complete solution concept. For instance: How can the air column be transformed into conductor when lightning discharges? How can the conductor be made to disappear immediately after the lightning has discharged?

ATTENTION!

When solving problem using ARIZ, the solution concept develops slowly (gradually).

It is risky to interrupt the problem solving process when an idea first appears or you may later find yourself fixing a half-developed idea. Carry the solving process through to the end of ARIZ!

3.4. identify the physical contradiction for the micro-level

Identify and describe the Physical Contradiction at **micro-level** using the following pattern:

there should be particles of a substance <indicate their physical state or action> in the Operational Zone in order to provide <indicate the macro-state according to step 3.3> and there should not be the particles (or particles should have the opposite state or action) in order to provide <indicate another macro-state according to step 3.3>

Example:

There should be <free charges> in the Air Column (when a lightning strikes) to provide <electrical conductivity to "remove" the lightning> and there should not be <free charges> (when the rest time)

to provide <prevent absorption of the radio waves>.

Comments:

27. It is not necessary in step 3.4 to precisely define the term "particles". For instance, the domains, molecules, ions etc. can all be considered as particles.

28. The particles may be:

- a) particles of a substances;
- b) combination of particles and field, or
- c) "particles of a field" (seldom).

29. If the problem has solution only on the Macro-Level, there may be difficulties to formulate step 3.4 because definition PhC on micro-level provides additional information: the problem is solved for on macro-level. In other words, attempting to formulate the Physical Contradiction for the micro-level can prove beneficial, if only because it provides us with the additional information that the problem has to be solved at the macro-level.

ATTENTION!

The first three parts of ARIZ essentially change the initial problem. The step 3.5 summarized this change. By formulating the Ideal Final Result IFR-2, we obtain an entirely new problem, a physical one. From this point we have to focus on this new problem!

3.5. formulate ifr-2

Identify and describe the Ideal Final Result (IFR-2) using the following pattern:

The Operational Zone <indicate> has to provide <indicate the opposite macro- or micro-states>

itself during the Operational Time <to indicate it>.

Example:

The <neutral molecules in the air column> has to <transform themselves into free charges> during <the lightning strikes>, and <the free charges> has to <transform itself into the neutral molecules> <after the lightning strikes>.

The meaning of this new problem is as follows: for the duration of the lightning discharge the free charges should appear on their own inside the air column; in this case the column of ionized air acts as the lightning-conductor and "attracts" the lightning. Immediately after

discharge the free charges in the air column should, on their own, become neutral molecules again. To solve this problem a knowledge of middle-school physics is need.

3.6. apply the inventive standards to resolve physical contradiction

Check possibility to apply the Inventive Standards to solve the new Physical Problem that formulated as the IFR-2. If after doing this, the problem is still unsolved, go to Part 4. If problem is solved using Inventive Standards it is possible to go at Part 7, however it is recommended to continue analysis through Part 4 anyway.

**ANNEXE-V. DEFINITION DE LA REPRESENTATION
CENTREE-OBJET**

Class Hierarchy for ontologie Project

Modele

Parametre

Fonction

Vepole

Interaction

Contradiction

Resultat-Ideal-Final

Etude

Ressources

Substance

Champ

Systeme

Espace

Temps

Generated on Tue Mar 09 11:40:37 CET 2004

Class Modele

Abstract Class Extends

TRIZ

Direct Instances:

None

Direct Subclasses:

- ?? Parametre
- ?? Fonction
- ?? Vepole
- ?? Interaction
- ?? Contradiction
- ?? Resultat-Ideal-Final

Class Documentation:

Elements de construction de la representation des problemes. Ce sont des points de vue de la realite.

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
considere		Boolean		0:1	
role		String		0:1	

Class Parametre

Concrete Class Extends

Modele

Direct Instances:

?? [devisabilite]

?? [La fixation]

?? [la position]

Direct Subclasses:

None

Class Documentation:

Element de description des ressources.

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
considere		Boolean		0:1	
valeur_maxi		Float		0:1	
valeur_mini		Float		0:1	
role		String		0:1	

Class Fonction

Concrete Class Extends

Modele

Direct Instances:

- ?? [dévissement impromptu]
- ?? [l'interchangeabilité du bijou]
- ?? [maintien du boîtier]
- ?? [maintien en position du boîtier de montre]
- ?? [mise en position du boîtier]
- ?? [reglage des ergots de maintien]
- ?? [Tenir le piercing sur la langue]

Direct Subclasses:

None

Class Documentation:

la fonction est realisee par un SYSTEME. la fonction porte sur un PRODUIT. la fonction peut etre d'IMPORTANTANCE PRINCIPALE ou TECHNIQUE la fonction peut etre de type UTILE ou NEFASTE. Elle peut participer a la realisation d'une ou plusieurs FONCTION-MERE.

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
valeur_initiale		String		0:1	
type		Symbol	Utile, Nefaste	0:1	
considere		Boolean		0:1	
valeur_objectif		String		0:1	
systeme		Instance	Systeme	0:1	

valeur_finale		String		0:1	
fonction-mere		Instance	Fonction	0:*	
produit		Instance	Substance, Systeme	0:1	
importance		Symbol	Principale, Technique	0:1	
role		String		0:1	
parametre		Instance	Parametre	0:*	

Class Vepole

Concrete Class Extends

Modele

Direct Instances:

None

Direct Subclasses:

None

Class Documentation:

Modele Substance-Champ de la TRIZ. Le Vepole est compose d'ELEMENTS qui generent des INTERACTIONS. il est caracterise par sa COMPLETUDE, la NECESSITE_DE-CONTACT entre les elements, les possibilites d'AJOUT_SUBSTANCE ou AJOUT_ADDITIF.

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
completude		Boolean		0:1	
fonctionnalite		Boolean		0:1	
necessite_de_contact		Boolean		0:1	
ajout_additif		Boolean		0:1	
considere		Boolean		0:1	
interactions		Instance	Interaction	0:*	
ajout_substance		Boolean		0:1	
elements		Instance	Systeme, Substance, Champ	0:*	

role		String		0:1	
ajout_champ		Boolean		0:1	

Class Interaction

Concrete Class Extends

Modele

Direct Instances:

None

Direct Subclasses:

None

Class Documentation:

Une INTERACTION est l'action d'un CHAMP entre des substances/systemes. L'action de ce champ permet la realisation d'une FONCTION, opere pendant TEMPS_OPERATIONNEL et se situe dans ESPACE_OPERATIONNEL. Elle peut etre de CARACTERE : INSUFFISANTE SATISFAISANTE EXCESSIVE ou NEFASTE.

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
caractere		Symbol	Insuffisante, Satisfaisante, Excessive, Nefaste	0:1	
temps_operationnel		Instance	Temps	0:1	
considere		Boolean		0:1	
champ		Instance	Champ	0:1	
role		String		0:1	
fonction		Instance	Fonction	0:1	
espace_operationnel		Instance	Espace	0:1	

Class Interaction

Concrete Class Extends

Modele

Direct Instances:

None

Direct Subclasses:

None

Class Documentation:

Une INTERACTION est l'action d'un CHAMP entre des substances/systemes. L'action de ce champ permet la realisation d'une FONCTION, opere pendant TEMPS_OPERATIONNEL et se situe dans ESPACE_OPERATIONNEL. Elle peut etre de CARACTERE : INSUFFISANTE SATISFAISANTE EXCESSIVE ou NEFASTE.

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
<i>caractere</i>		Symbol	Insuffisante, Satisfaisante, Excessive, Nefaste	0:1	
<i>temps_operationnel</i>		Instance	Temps	0:1	
<i>considere</i>		Boolean		0:1	
<i>champ</i>		Instance	Champ	0:1	
<i>role</i>		String		0:1	
<i>fonction</i>		Instance	Fonction	0:1	
<i>espace_operationnel</i>		Instance	Espace	0:1	

Class Contradiction

Concrete Class Extends

Modele

Direct Instances:

?? [contradiction]

?? [contradiction_outil]

Direct Subclasses:

None

Class Documentation:

Modele CONTRADICTION de la TRIZ. Opposition entre VALEUR1 et VALEUR2 requises pour le PARAMETRE, de l'ELEMENTS, afin de realiser respectivement FONCTION1 durant TEMPS1 et a ESPACE1 et FONCTION2 durant TEMPS2 et a ESPACE2.

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
temps2		Instance	Temps	0:1	
valeur1		String		0:1	
espace2		Instance	Espace	0:1	
considere		Boolean		0:1	
elements		Instance	Systeme, Substance	0:*	
valeur2		String		0:1	
temps1		Instance	Temps	0:1	
fonction2		Instance	Fonction	0:1	

espace1		Instance	Espace	0:1	
fonction1		Instance	Fonction	0:1	
role		String		0:1	
parametre		Instance	Parametre	0:*	

Class Resultat-Ideal-Final

Concrete Class Extends

Modele

Direct Instances:

?? [RIF]

Direct Subclasses:

None

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
temps_operationnel		Instance	Temps	0:1	
considere		Boolean		0:1	
fonction_utile		Instance	Fonction	0:1	
role		String		0:1	
fonction_nefaste		Instance	Fonction	0:1	
espace_operationnel		Instance	Espace	0:1	

Class Etude

Concrete Class Extends

TRIZ

Direct Instances:

?? [potence horlogère]

Direct Subclasses:

None

Class Documentation:

Projet en cours, dont on fixe l'OBJECTIF en fonction du mode d'accroissement de l'idealite.

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
<i>considere</i>		Boolean		0:1	
<i>objectif</i>		String		0:1	
<i>role</i>		String		0:1	

Class Ressources

Concrete Class Extends

TRIZ

Direct Instances:

None

Direct Subclasses:

?? Substance
 ?? Champ
 ?? Systeme
 ?? Espace
 ?? Temps

Class Documentation:

Toute ressource sera potentiellement utile lors de la resolution. Elles sont decrites par des PARAMETRES. Elles ont une LOCALISATION.

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
considere		Boolean		0:1	
role		String		0:1	
localisation		String		0:1	
parametre		Instance	Parametre	0:*	

Class Substance

Concrete Class Extends

Ressources

Direct Instances:

?? [Le bijou]
 ?? [le boitier]
 ?? [les ergots]

Direct Subclasses:

None

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
considere		Boolean		0:1	
role		String		0:1	
localisation		String		0:1	
parametre		Instance	Parametre	0:*	

Class Champ

Concrete Class Extends

Ressources

Direct Instances:

None

Direct Subclasses:

None

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
nature		Symbol	Mecanique, Acoustique, Chimique, Thermique, Electrique, Magnetique	0:1	
considere		Boolean		0:1	
role		String		0:1	
localisation		String		0:1	
parametre		Instance	Parametre	0:*	

Class Systeme

Concrete Class Extends

Ressources

Direct Instances:

- ?? [Système de fixation piercing]
- ?? [système de maintien en position de la montre]

Direct Subclasses:

None

Class Documentation:

Description du systeme selon la loi 1 des lois d'evolution de la TRIZ. Le systeme est compose d'un MOTEUR transformant l'energie exterieure en energie utilisable par le systeme; d'un OUTIL qui agit directement sur le produit; d'une TRANSMISSION qui transporte l'enrgie du moteur a l'outil et d'un CONTROLE, qui adapte la realisation de la fonction en fonction de l'environnement.

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
moteur		Instance	Substance, Systeme	0:1	
considere		Boolean		0:1	
controle		Instance	Substance, Systeme	0:1	
outil		Instance	Substance, Systeme	0:1	
role		String		0:1	

localisation		String		0:1	
parametre		Instance	Parametre	0:*	
transmission		Instance	Substance, Systeme	0:1	

Class Espace

Concrete Class Extends

Ressources

Direct Instances:

None

Direct Subclasses:

None

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
considere		Boolean		0:1	
role		String		0:1	
localisation		String		0:1	
parametre		Instance	Parametre	0:*	

Class Temps

Concrete Class Extends

Ressources

Direct Instances:

None

Direct Subclasses:

None

Template Slots

Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
origine		String		0:1	
considere		Boolean		0:1	
role		String		0:1	
localisation		String		0:1	
parametre		Instance	Parametre	0:*	

ANNEXE-VI. ECRITURE DU MODELE DE CONNAISSANCES EN LOGIQUE DE DESCRIPTION

:: Knowledge Base Definition

:: Author: Alexis Bultey

:: version: CICLOP 2

:: version 0.2:23/04/2004

(newKB triz2)

(newtbox triz)

:: ROLES

:: =====

(define-atomic-role parametre : domain (OR Fonction Contradiction Ressources): range
Parametre)

(define-atomic-role considere :feature true : domain Triz : range Boolean)

(define-atomic-role role :feature true : domain Triz : range Boolean)

(define-atomic-role localisation :feature true : domain Ressources: range Symbol)

(define-atomic-role systeme :feature true : domain Fonction : range Systeme)

(define-atomic-role nature : feature true : domain Champ : range Symbol)

(define-atomic-role origine : feature true : domain Temps: range Float)

(define-atomic-role moteur : feature true : domain Systeme : range (OR Substance Systeme))

(define-atomic-role outil : feature true : domain Systeme : range (OR Substance Systeme))

(define-atomic-role controle : feature true: domain Systeme : range (OR Substance Systeme))

(define-atomic-role transmission : feature true : domain Systeme : range (OR Substance Systeme))

(define-atomic-role objectif : feature true : domain Etude : range String)

(define-atomic-role valeur_initiale : feature true : domain Fonction :range String)

(define-atomic-role type : feature true : domain Fonction : range Symbol)

(define-atomic-role valeur_objectif : feature true : domain Fonction : range String)

(define-atomic-role valeur_finale : feature true : domain Fonction : range String)

(define-atomic-role fonction_mere : domain Fonction : range Fonction)

(define-atomic-role produit : feature true : domain Fonction : range (OR Substance Systeme))

(define-atomic-role importance : feature true : domain Fonction : range Symbol)

(define-atomic-role caractere : feature true : domain Interaction : range Symbol)

(define-atomic-role temps_operationnel : feature true : domain (OR Interaction RIF) :
range Temps)

(define-atomic-role champ : feature true : domain Interaction : range Champ)

(define-atomic-role fonction : feature true : domain Interaction : range Fonction)

(define-atomic-role espace_operationnel : feature true : domain (OR Interaction RIF):
range Espace)

(define-atomic-role completude : feature true : domain Vepole : range Boolean)

(define-atomic-role fonctionnalite : feature true : domain Vepole : range Boolean)

(define-atomic-role necessite_de_contact : feature true : domain Vepole : range Boolean)

(define-atomic-role ajout_additif : feature true : domain Vepole : range Boolean)

(define-atomic-role interactions: domain Vepole : range Interaction)

(define-atomic-role ajout_substance: feature true : domain Vepole : range Boolean)

(define-atomic-role elements : domain (OR Contradiction Vepole) : range (OR Systeme
Substance Champ))

(define-atomic-role ajout_champ : feature true : domain Vepole : range Boolean)

(define-atomic-role temps2 : feature true : domain Contradiction : range Temps)

(define-atomic-role valeur1 : feature true : domain Contradiction: range String)

(define-atomic-role espace2 : feature true : domain Contradiction : range Espace)

(define-atomic-role valeur2 : feature true : domain Contradiction : range String)

(define-atomic-role temps1 : feature true : domain Contradiction : range Temps)

(define-atomic-role fonction2 : feature true : domain Contradiction : range fonction)

(define-atomic-role espace1 : feature true : domain Contradiction : range Espace)

(define-atomic-role fonction1 : feature true : domain Contradiction : range Fonction)

(define-atomic-role fonction_utile : feature true : domain RIF : range Fonction)

(define-atomic-role fonction_nefaste : feature true : domain RIF : range fonction)

(define-atomic-role valeur_mini : feature true : domain Parametre : range Float)

(define-atomic-role valeur_maxi : feature true : domain Parametre : range Float)

:: CONCEPTS

:: =====

(define-atomic-concept Symbol)

(define-atomic-concept String)

(define-atomic-concept Float)

(define-atomic-concept Boolean)

(define-atomic-concept Triz)

(define-concept Modele

(AND TRIZ

(ALL considere Boolean)

(ALL role String)))

(define-concept Parametre

(AND Modele

(ALL valeur_maxi Float)

(ALL valeur_mini Float)))

(define-concept Ressources

(AND Triz

(ALL considere Boolean)

(ALL role String)

(ALL localisation String)

(ALL parametre Parametre)))

(define-concept Substance Ressources:primitive true)

(define-concept Nature_Champ Symbol:primitive true)

(define-concept Champ
 (AND Ressources
 (ALL nature Nature_Champ)))

(define-concept Espace Ressources : primitive true)

(define-concept Temps
 (AND Ressources
 (ALL origine String)))

(define-concept Systeme
 (AND Ressources
 (ALL moteur (OR Substance Systeme))
 (ALL transmission (OR Substance Systeme))
 (ALL outil (OR Substance Systeme))
 (ALL controle (OR Substance Systeme))))

(define-concept Etude
 (AND Triz
 (ALL role String)
 (ALL considere Boolean)
 (ALL objectif String)))

(define-concept Type_Fonction Symbol : primitive true)

(define-concept Utile
 (AND Type_Fonction
 (NOT Nefaste)))

(define-concept Nefaste
 (AND Type_Fonction
 (NOT Utile)))

(define-concept Importance_Fonction Symbol : primitive true)

(define-concept Principale
 (AND Importance_Fonction
 (NOT Technique)))

(define-concept Technique
 (AND Importance_Fonction
 (NOT Principale)))

(define-concept Fonction
 (AND Modele
 (ALL valeur_initiale String)
 (ALL type Type_Fonction)
 (ALL valeur_objectif String)
 (ALL systeme Systeme)

(ALL valeur_finale String)

(ALL fonction_mere Fonction)

(ALL produit (OR Substance Champ))

(ALL importance Importance_Fonction)

(ALL parametre Parametre)))

(define-concept Caractere_Interaction Symbol:primitive true)

(define-concept Interaction

(AND Modele

(ALL caractere Caractere_Interaction)

(ALL temps_operationnel Temps)

(ALL champ Champ)

(ALL fonction Fonction)

(ALL espace_operationnel Espace)))

(define-concept Vepole

(AND Modele

(ALL completude Boolean)

(ALL fonctionnalite Boolean)

(ALL necessite_de_contact Boolean)

(ALL ajout_additif Boolean)

(ALL interactions Interaction)

(ALL ajout_substance Boolean)

(ALL elements (OR Systeme Champ Substance))

(ALL ajout_champ Boolean)))

(define-concept Contradiction

(AND Modele

(ALL temps2 Temps)

(ALL valeur1 String)

(ALL espace2 Espace)

(ALL elements (OR Systeme Substance))

(ALL valeur2 String)

(ALL temps1 Temps)

(ALL fonction2 Fonction)

(ALL espace1 Espace)

(ALL fonction1 Fonction)

(ALL parametre Parametre)))

(define-concept RIF

(AND Modele

(ALL temps_operationnel Temps)

(ALL fonction_utile Fonction)

(ALL fonction_nefaste Fonction)

(ALL espace_operationnel Espace)))

(classify-all)

(display-graphical-classification)

(ClassifyAndAnalyse)

(inspect)

(GenerateHTML "c:\\transfert\\genHTML" :tbox \$CURRENT\$)

ANNEXE-VII. RÈGLES DÉFINIES DANS JESS

```
(defrule INITIALISE::bind_index
  (MAIN::object (is-a Reponse) (index ?index&:(> ?index ?*index*)))
  =>
  (if (> ?index ?*index*) then (bind ?*index* ?index)))
```

```
(defrule INTERFACE::autorise_multi
  (declare (saliency 100))
  (MAIN::object (is-a Reponse) (reponse ?reponse) (numero ?num))
  ?n <- (MAIN::object (is-a Autorisation) (:NAME ?nom) (reponse ?reponse&:(neq
?reponse nil)) (numero ?num))
  =>
  (if (neq (length$ (fact-slot-value (fact-id ?n) conditions)) 0) then (return))
  (bind ?autorise (slot-get (str-cat ?nom) autorise))
  (foreach ?item ?autorise (make-instance ?item of Pose) (slot-set (str-cat ?item) numero
?item)))
```

```
(defrule INTERFACE::autorise_simple
  (declare (saliency 100))
  (MAIN::object (is-a Reponse) (numero ?num))
  ?n <- (MAIN::object (is-a Autorisation) (:NAME ?nom) (reponse nil) (numero ?num))
  =>
  (if (neq (length$ (fact-slot-value (fact-id ?n) conditions)) 0) then (return))
  (bind ?autorise (slot-get (str-cat ?nom) autorise))
  (foreach ?item ?autorise (make-instance ?item of Pose) (slot-set (str-cat ?item) numero
?item)))
```

```
(defrule INTERFACE::autorise_une_condition
  (declare (saliency 100))
  (MAIN::object (is-a Reponse) (numero ?num))
  (MAIN::object (is-a Autorisation) (:NAME ?nom) (reponse nil) (numero ?num)
  (conditions ?condition1) (autorise ?autorise))
  (MAIN::object (is-a Reponse) (numero ?condition1))
  =>
  (bind ?autorise (slot-get (str-cat ?nom) autorise))
  (foreach ?item ?autorise (make-instance ?item of Pose) (slot-set (str-cat ?item) numero
  ?item)))
```

```
(defrule INTERFACE::cree_parametre
  ?objet <- (MAIN::object (parametre ?parametre&:(neq ?parametre nil)))
  (not (MAIN::object (is-a Parametre) (OBJECT ?parametre)))
  =>
  (make-instance ?parametre of Parametre)
  (slot-set ?parametre :NAME (str-cat ?parametre)))
```

```
(defrule INTERFACE::fonction_nefaste
  (MAIN::object (OBJECT ?systeme) (role "systeme_nefaste"))
  (MAIN::object (is-a Fonction) (role ?role&:(neq ?role "fonction_nefaste")) (:NAME
  ?fonction) (systeme ?systeme))
  =>
  (slot-set ?fonction role "fonction_nefaste"))
```

```
(defrule INTERFACE::fonction_utile
  (MAIN::object (OBJECT ?systeme) (role "systeme_utile"))
```

```
(MAIN::object (is-a Fonction) (role ?role&:(neq ?role "fonction_utile")) (:NAME
?fonction) (systeme ?systeme))
```

```
=>
```

```
(slot-set ?fonction role "fonction_utile"))
```

```
(defrule INTERFACE::modifie
```

```
  ?modif <- (REPONSE::modifie (num ?num) (slot ?slot))
```

```
(MAIN::object (is-a Reponse) (reponse ?reponse) (numero ?num))
```

```
=>
```

```
(if (eq (or (eq (str-cat (slot-cardinality (class ?reponse) ?slot)) "1 +oo") (eq (str-cat (slot-
cardinality (class ?reponse) ?slot)) "0 +oo")) TRUE) then (slot-insert$ ?reponse (instance-name
?slot) 1 ?*repondu*) else (slot-set ?reponse (instance-name ?slot) ?*repondu*))
```

```
(retract ?modif))
```

```
(defrule INTERFACE::systeme_nefaste
```

```
(MAIN::object (role "fonction_nefaste") (systeme ?systeme))
```

```
(MAIN::object (is-a Systeme) (OBJECT ?systeme) (role ?role&:(neq ?role
"systeme_nefaste"))))
```

```
=>
```

```
(slot-set ?systeme role "systeme_nefaste"))
```

```
(defrule INTERFACE::systeme_utile
```

```
(MAIN::object (role "fonction_utile") (systeme ?systeme))
```

```
(MAIN::object (is-a Systeme) (OBJECT ?systeme) (role ?role&:(neq ?role
"systeme_utile"))))
```

```
=>
```

```
(slot-set ?systeme role "systeme_utile"))
```

```

(defrule INTERFACE::vepole-interaction
  (MAIN::object (is-a Vepole) (:NAME ?n) (elements $?w ?c&:(element-est-un Champ
?c) $?x))
  (MAIN::object (is-a Vepole) (:NAME ?n) (interactions $?a ?interaction $?b))
  (not (MAIN::object (is-a Interaction) (champ ?c)))
  =>
  (make-instance ?interaction of Interaction)
  (slot-set ?interaction :NAME (str-cat ?interaction))
  (slot-set ?interaction champ ?c))

(defrule MAIN::Vepole_complet
  ?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (elements $?a ?s1&:(element-est-
un Substance ?s1) ?s2&:(element-est-un Substance ?s2) $?b))
  (MAIN::object (is-a Vepole) (:NAME ?n) (elements $?w ?c&:(element-est-un Champ
?c) $?x))
  (not (MAIN::object (is-a Vepole) (:NAME ?n) (completude TRUE)))
  =>
  (slot-set ?vepole completude TRUE))

(defrule MAIN::Vepole_fonctionnel
  ?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (fonctionnalite ?fonctionnel&:(neq
?fonctionnel TRUE)) (completude TRUE) (interactions $?inter1 ?objet $?inter2))
  (MAIN::object (is-a Interaction) (OBJECT ?objet) (fonction ?fonction) (caractere
?caractere&:(eq (str-cat ?caractere) "Satisfaisante")))
  (MAIN::object (is-a Fonction) (OBJECT ?fonction) (type ?type&:(eq (str-cat ?type)
"Utile")))
  =>
  (slot-set ?vepole fonctionnalite TRUE))

```

```

(defrule MAIN::Vepole_incomplet
  ?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n))

  (not (and (MAIN::object (is-a Vepole) (:NAME ?n) (elements $?_2_a
?_2_s1&:(element-est-un Substance ?_2_s1) ?_2_s2&:(element-est-un Substance ?_2_s2)
$_2_b)) (MAIN::object (is-a Vepole) (:NAME ?n) (elements $_2_w ?_2_c&:(element-est-un
Champ ?_2_c) $_2_x))))

  (not (MAIN::object (is-a Vepole) (:NAME ?n) (completude FALSE)))

=>

  (slot-set ?vepole completude FALSE))

(defrule MAIN::Vepole_infonctionnel
  ?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (completude FALSE))

  (not (MAIN::object (is-a Vepole) (:NAME ?n) (fonctionnalite FALSE)))

=>

  (slot-set ?vepole fonctionnalite FALSE))

(defrule MAIN::applique_regle
  (MAIN::object (is-a Regle) (numero ?num) (objet ?objet) (slot ?slot) (valeur
?valeur&:(not (integerp (str-index reponse ?valeur))))))

  (MAIN::object (is-a Reponse) (numero ?num))

  (MAIN::object (is-a Reponse) (reponse ?reponse) (numero ?objet))

=>

  (bind ?slot (slot-get ?slot :NAME))

  (if (eq (or (eq (str-cat (slot-cardinality (class ?reponse) ?slot)) "1 +oo") (eq (str-cat (slot-
cardinality (class ?reponse) ?slot)) "0 +oo")) TRUE) then (slot-insert$ ?reponse (instance-name
?slot) 1 ?valeur) else (slot-set ?reponse (instance-name ?slot) ?valeur)))

```

```
(defrule MAIN::applique_regle_2
  (MAIN::object (is-a Regle) (numero ?num&:(eq (str-cat ?num) (str-cat ?objet))) (objet
?objet) (slot ?slot) (valeur ?valeur&:(not (integerp (str-index reponse ?valeur))))))
  (MAIN::object (is-a Reponse) (reponse ?reponse) (numero ?num))
  =>
  (bind ?slot (slot-get ?slot :NAME))
  (if (eq (or (eq (str-cat (slot-cardinality (class ?reponse) ?slot)) "1 +oo") (eq (str-cat (slot-
cardinality (class ?reponse) ?slot)) "0 +oo"))) TRUE) then (slot-insert$ ?reponse (instance-name
?slot) 1 ?valeur) else (slot-set ?reponse (instance-name ?slot) ?valeur)))
```

```
(defrule MAIN::applique_regle_question
  (MAIN::object (is-a Regle) (numero ?num) (objet ?objet) (slot ?slot) (valeur ?question))
  (MAIN::object (is-a Reponse) (numero ?num))
  (MAIN::object (is-a Reponse) (reponse ?reponse) (numero ?objet))
  (MAIN::object (is-a Reponse) (reutilisation ?question) (reponse ?valeur))
  =>
  (bind ?slot (slot-get ?slot :NAME))
  (if (eq (or (eq (str-cat (slot-cardinality (class ?reponse) ?slot)) "1 +oo") (eq (str-cat (slot-
cardinality (class ?reponse) ?slot)) "0 +oo"))) TRUE) then (slot-insert$ ?reponse (instance-name
?slot) 1 ?valeur) else (slot-set ?reponse (instance-name ?slot) ?valeur)))
```

```
(defrule MAIN::applique_regle_question_2
  (MAIN::object (is-a Regle) (numero ?num) (objet ?objet) (slot ?slot) (valeur ?question))
  (MAIN::object (is-a Reponse) (numero ?num))
  (MAIN::object (is-a Reponse) (reutilisation ?question) (reponse ?valeur))
  =>
  (bind ?slot (slot-get ?slot :NAME))
```

```
(if (eq (or (eq (str-cat (slot-cardinality (class ?objet) ?slot)) "1 +oo") (eq (str-cat (slot-cardinality (class ?objet) ?slot)) "0 +oo"))) TRUE) then (slot-insert$ ?objet (instance-name ?slot) 1 ?valeur) else (slot-set ?objet (instance-name ?slot) ?valeur)))
```

```
(defrule MAIN::applique_regle_role
```

```
(MAIN::object (is-a Regle) (numero ?num) (objet ?objet&:(not (integerp (str-index reponse ?objet)))) (slot ?slot) (valeur ?valeur&:(not (integerp (str-index reponse ?valeur))))))
```

```
(MAIN::object (is-a Reponse) (numero ?num))
```

```
(MAIN::object (role ?objet) (:NAME ?nom))
```

```
=>
```

```
(if (eq (or (eq (str-cat (slot-cardinality (class ?nom) ?slot)) "1 +oo") (eq (str-cat (slot-cardinality (class ?nom) ?slot)) "0 +oo"))) TRUE) then (slot-insert$ ?nom (instance-name ?slot) 1 ?valeur) else (slot-set ?nom (instance-name ?slot) ?valeur)))
```

```
(defrule MAIN::applique_regle_role_2
```

```
(MAIN::object (is-a Regle) (numero ?num) (objet ?objet) (slot ?slot) (valeur ?question))
```

```
(MAIN::object (is-a Reponse) (numero ?num))
```

```
(MAIN::object (is-a Reponse) (reutilisation ?question) (reponse ?valeur))
```

```
(MAIN::object (role ?objet) (:NAME ?nom))
```

```
=>
```

```
(if (eq (or (eq (str-cat (slot-cardinality (class ?nom) ?slot)) "1 +oo") (eq (str-cat (slot-cardinality (class ?nom) ?slot)) "0 +oo"))) TRUE) then (slot-insert$ ?nom (instance-name ?slot) 1 ?valeur) else (slot-set ?nom (instance-name ?slot) ?valeur)))
```

```
(defquery MAIN::cherche_objets
```

```
(declare (variables ?objet))
```

```
(MAIN::object (OBJECT ?objet)))
```

```
(defquery MAIN::cherche_poses
```

```
(MAIN::object (is-a Pose)))
```

```
(defquery MAIN::cherche_reponse  
  (declare (variables ?reutilise))  
  (MAIN::object (is-a Reponse) (reutilisation ?reutilise)))
```

```
(defquery MAIN::cherche_role  
  (declare (variables ?role))  
  (MAIN::object (role ?role)))
```

```
(defquery MAIN::liste_valides  
  (declare (variables ?type))  
  (MAIN::object (is-a ?type)))
```

```
(defquery MAIN::question_precedente  
  (declare (variables ?index))  
  (MAIN::object (is-a Reponse) (index ?index)))
```

```
(defrule QUESTION::pose_question  
  (declare (salience 500))  
  (MAIN::object (is-a Question) (numero ?num) (classe ?classe) (objet ?objet) (action  
?action) (texte ?texte) (type_reponse ?type) (slot ?slot) (valide $?valide) (valide_incremente  
?incremente))  
  ?q <- (MAIN::object (is-a Pose) (numero ?num))  
  =>  
  (printout t ?num crlf ?type crlf ?action crlf)  
  (bind ?texte (valide_texte ?texte))  
  (call ?*fenetrequestion* removeAll)
```

```

(call ?*fenetrequestion* validate)

(if (and (eq ?incremente TRUE) (eq (length$ $?valide) 0)) then (bind $?valide
(liste_valides (slot-get ?classe :NAME))))

(pose_question ?texte ?type $?valide)

(if (neq ?*quitter* TRUE) then (if (eq ?action rien) then (bind ?*repondu* "")) (if (eq
?action modifie) then (if (eq (str-cat ?objet) "nil") then (bind ?objet (slot-get ?classe :NAME)))
(modifie ?objet ?slot)) (unmake-instance ?q) (cree_reponse ?num ?type) (clear-focus-stack)
(focus MAIN) (run) (focus INTERFACE) (run) (focus QUESTION) (run) (return))

(if (eq ?*precedent* TRUE) then (bind ?*precedent* FALSE) (question-precedente))

(return))

```

```

(defrule QUESTION::question_parametre

(declare (saliency 200))

(MAIN::object (is-a Fonction) (:NAME ?name) (considere TRUE) (produit ?objet&:(neq
objet nil)) (parametre nil))

(MAIN::object (OBJECT ?objet) (:NAME ?produit))

=>

(bind ?question (str-cat (create$ ?name " est la modification d'un parametre en particulier
de " ?produit ". Quel est ce parametre ?")))

(pose_question ?question multiouvert (liste_valides Parametre))

(while (neq ?*quitter* TRUE) (if (not (instance-existp (str-cat ?*repondu*))) then (make-
instance (str-cat ?*repondu*) of Parametre) (slot-set (str-cat ?*repondu*) :NAME (str-cat
?*repondu*)) (slot-set ?name parametre (str-cat ?*repondu*)) (cree_reponse 501 simple) (clear-
focus-stack) (focus MAIN) (run) (focus INTERFACE) (run) (focus QUESTION) (run) (return)))

```

```

(defrule QUESTION::question_produit

(declare (saliency 200))

(MAIN::object (is-a Fonction) (:NAME ?name) (considere TRUE) (produit nil))

=>

```

```
(bind ?question (str-cat (create$ "La realisation d'une fonction s'effectue dans le but de
modifier une caracteristique d'un objet. Quel est l'objet en question, pour la fonction " ?name "
?"))))
```

```
(pose_question ?question multiouvert (liste_valides Substance))
```

```
(if (neq ?*quitter* TRUE) then (slot-set ?name produit ?*repondu*) (cree_reponse 502
simple) (clear-focus-stack) (focus MAIN) (run) (focus INTERFACE) (run) (focus QUESTION)
(run) (return)))
```

```
(defrule QUESTION::question_systeme_controle
```

```
(declare (saliency 180))
```

```
(MAIN::object (is-a Systeme) (OBJECT ?objet) (role ?role) (:NAME ?nom) (considere
TRUE) (controle nil))
```

```
(MAIN::object (is-a Fonction) (:NAME ?fonction) (produit ?objetproduit) (systeme
?objet))
```

```
(MAIN::object (OBJECT ?objetproduit) (:NAME ?produit))
```

```
=>
```

```
(bind ?question (str-cat (create$ "Si il est necessaire de moduler la realisation de "
?fonction ". Un element de " ?nom " assure cette action de controle. Quel est l'element en
question ?"))))
```

```
(pose_question ?question multiouvert (liste_valides Substance))
```

```
(while (neq ?*quitter* TRUE) (slot-set ?nom controle ?*repondu*) (if (eq ?role
"systeme_utile") then (slot-set ?*repondu* role "controle_utile")) (if (eq ?role
"systeme_nefaste") then (slot-set ?*repondu* role "controle_nefaste")) (cree_reponse 503
simple) (clear-focus-stack) (focus MAIN) (run) (focus INTERFACE) (run) (focus QUESTION)
(run) (return)))
```

```
(defrule QUESTION::question_systeme_moteur
```

```
(declare (saliency 180))
```

```
(MAIN::object (is-a Systeme) (OBJECT ?objet) (role ?role) (:NAME ?nom) (considere
TRUE) (moteur nil))
```

```
(MAIN::object (is-a Fonction) (:NAME ?fonction) (produit ?objetproduit) (systeme ?objet))
```

```
(MAIN::object (OBJECT ?objetproduit) (:NAME ?produit))
```

```
=>
```

```
(bind ?question (str-cat (create$ "Lors de la realisation de " ?fonction ", un element de " ?nom " transforme l'energie exterieure en energie utilisable par " ?nom ". Quel est l'element en question ?")))
```

```
(pose_question ?question multiouvert (liste_valides Substance))
```

```
(while (neq ?*quitter* TRUE) (slot-set ?nom moteur ?*repondu*) (if (eq ?role "systeme_utile") then (slot-set ?*repondu* role "moteur_utile")) (if (eq ?role "systeme_nefaste") then (slot-set ?*repondu* role "moteur_nefaste"))) (cree_reponse 504 simple) (clear-focus-stack) (focus MAIN) (run) (focus INTERFACE) (run) (focus QUESTION) (run) (return)))
```

```
(defrule QUESTION::question_systeme_outil
```

```
(declare (saliency 190))
```

```
(MAIN::object (is-a Systeme) (OBJECT ?objet) (:NAME ?nom) (considere TRUE) (outil nil))
```

```
(MAIN::object (is-a Fonction) (:NAME ?fonction) (produit ?produit) (systeme ?objet))
```

```
=>
```

```
(bind ?question (str-cat (create$ "Lors de la realisation de " ?fonction ", un element de " ?nom " est en contact direct, agit directement sur " ?produit ". Quel est l'element en question ?")))
```

```
(pose_question ?question multiouvert (liste_valides Substance))
```

```
(while (neq ?*quitter* TRUE) (slot-set ?nom outil ?*repondu*) (if (eq ?role "systeme_utile") then (slot-set ?*repondu* role "outil_utile")) (if (eq ?role "systeme_nefaste") then (slot-set ?*repondu* role "outil_nefaste"))) (cree_reponse 505 simple) (clear-focus-stack) (focus MAIN) (run) (focus INTERFACE) (run) (focus QUESTION) (run) (return)))
```

```
(defrule QUESTION::question_systeme_sans_fonction
```

```
(declare (saliency 600))
```

```
?systeme <- (MAIN::object (is-a Systeme) (OBJECT ?objet) (:NAME ?nom) (considere
TRUE))
```

```
(not (MAIN::object (is-a Fonction) (systeme ?objet)))
```

```
=>
```

```
(bind ?question (str-cat (create$ "Quelle est la fonction principale realisee par " ?nom "
?"))))
```

```
(printout t test1)
```

```
(pose_question ?question multiouvert (liste_valides Fonction))
```

```
(printout t test2)
```

```
(while (neq ?*quitter* TRUE) (if (not (instance-existp (str-cat ?*repondu*))) then (make-
instance (str-cat ?*repondu*) of Fonction) (slot-set (str-cat ?*repondu*) :NAME (str-cat
?*repondu*)) (slot-set (str-cat ?*repondu*) importance Principale) (slot-set (str-cat ?*repondu*)
systeme ?systeme) (cree_reponse 506 simple) (clear-focus-stack) (focus MAIN) (run) (focus
INTERFACE) (run) (focus QUESTION) (run) (return)))
```

```
(defrule QUESTION::question_systeme_transmission
```

```
(declare (saliency 170))
```

```
(MAIN::object (is-a Systeme) (OBJECT ?objet) (:NAME ?nom) (considere TRUE)
(outil ?outil) (moteur ?moteur))
```

```
(MAIN::object (is-a Fonction) (:NAME ?fonction) (systeme ?objet))
```

```
=>
```

```
(bind ?question (str-cat (create$ "Lors de la realisation de " ?fonction ", un element de "
?nom " transporte l'energie delivree par " ?moteur "jusqu'a " ?outil ". Quel est l'element en
question ?"))))
```

```
(pose_question ?question multiouvert (liste_valides Substance))
```

```
(while (neq ?*quitter* TRUE) (slot-set ?nom transmission ?*repondu*) (if (eq ?role
"systeme_utile") then (slot-set ?*repondu* role "transmission_utile")) (if (eq ?role
"systeme_nefaste") then (slot-set ?*repondu* role "transmission_nefaste")) (cree_reponse 507
simple) (clear-focus-stack) (focus MAIN) (run) (focus INTERFACE) (run) (focus QUESTION)
(run) (return)))
```

```
(defrule QUESTION::question_valeurfinale
  (declare (salience 200))
  (MAIN::object (is-a Fonction) (:NAME ?name) (considere TRUE) (produit ?produit)
(valeur_initiale ?valeur_initiale&:(neq ?valeur_initiale nil)) (parametre ?parametre)
(valeur_finale nil))
  =>
  (bind ?parametre (slot-get ?parametre :NAME))
  (bind ?question (str-cat (create$ "Quelle est la valeur de " ?parametre " apres l'execution
de " ?name " ?")))
  (pose_question ?question simple nil)
  (while (neq ?*quitter* TRUE) (slot-set ?name valeur_finale ?*repondu*) (cree_reponse
508 simple) (clear-focus-stack) (focus MAIN) (run) (focus INTERFACE) (run) (focus
QUESTION) (run) (return)))
```

```
(defrule QUESTION::question_valeurinitiale
  (declare (salience 200))
  (MAIN::object (is-a Fonction) (:NAME ?name) (considere TRUE) (produit ?produit)
(valeur_initiale nil) (parametre ?parametre&:(neq ?parametre nil)))
  =>
  (bind ?parametre (slot-get ?parametre :NAME))
  (bind ?question (str-cat (create$ "Quelle est la valeur de " ?parametre " avant l'execution
de " ?name " ?")))
  (pose_question ?question simple nil)
  (while (neq ?*quitter* TRUE) (slot-set ?name valeur_initiale ?*repondu*) (cree_reponse
509 simple) (clear-focus-stack) (focus MAIN) (run) (focus INTERFACE) (run) (focus
QUESTION) (run) (return)))
```

```
(defrule RESET_DOMAINE::reset_champ
```

```
(MAIN::object (is-a Champ) (:NAME ?nom))
```

```
=>
```

```
(unmake-instance ?nom))
```

```
(defrule RESET_DOMAINE::reset_espace
```

```
(MAIN::object (is-a Espace) (:NAME ?nom))
```

```
=>
```

```
(unmake-instance ?nom))
```

```
(defrule RESET_DOMAINE::reset_fonction
```

```
(MAIN::object (is-a Fonction) (:NAME ?nom))
```

```
=>
```

```
(unmake-instance ?nom))
```

```
(defrule RESET_DOMAINE::reset_parametre
```

```
(MAIN::object (is-a Parametre) (:NAME ?nom))
```

```
=>
```

```
(unmake-instance ?nom))
```

```
(defrule RESET_DOMAINE::reset_substance
```

```
(MAIN::object (is-a Substance) (:NAME ?nom))
```

```
=>
```

```
(unmake-instance ?nom))
```

```
(defrule RESET_DOMAINE::reset_systeme
```

```
(MAIN::object (is-a Systeme) (:NAME ?nom))
```

```
=>
```

```
(unmake-instance ?nom))
```

```
(defrule RESET_DOMAINE::reset_temps  
  (MAIN::object (is-a Temps) (:NAME ?nom))  
  =>  
  (unmake-instance ?nom))
```

```
(defrule RESET_ETUDE::initialise_considere  
  ?objet <- (MAIN::object (considere ?val&:(neq ?val FALSE)))  
  =>  
  (slot-set ?objet considere FALSE))
```

```
(defrule RESET_ETUDE::initialise_contradiction  
  (declare (saliency 100))  
  (MAIN::initial-fact)  
  =>  
  (initialize-instance contradiction)  
  (slot-set contradiction role contradiction)  
  (initialize-instance contradiction_outil)  
  (slot-set contradiction_outil role contradiction_outil))
```

```
(defrule RESET_ETUDE::initialise_etudes  
  (declare (saliency 100))  
  (MAIN::object (is-a Etude) (:NAME ?nom))  
  =>  
  (unmake-instance ?nom))
```

```
(defrule RESET_ETUDE::initialise_interactions
  (declare (saliency 100))
  (MAIN::object (is-a Interaction) (:NAME ?nom))
  =>
  (unmake-instance ?nom))
```

```
(defrule RESET_ETUDE::initialise_pose
  (declare (saliency 100))
  (MAIN::object (is-a Pose) (:NAME ?nom))
  =>
  (unmake-instance ?nom))
```

```
(defrule RESET_ETUDE::initialise_reponses
  (declare (saliency 100))
  (MAIN::object (is-a Reponse) (:NAME ?nom))
  =>
  (unmake-instance ?nom))
```

```
(defrule RESET_ETUDE::initialise_rif
  (declare (saliency 100))
  (MAIN::initial-fact)
  =>
  (initialize-instance RIF)
  (slot-set RIF role RIF))
```

```
(defrule RESET_ETUDE::initialise_roles
  (MAIN::object (role ?role&:(neq ?role nil)) (:NAME ?nom))
```

=>

```
(slot-set ?nom role nil)
```

```
(defrule RESET_ETUDE::initialise_vepoles
```

```
  (declare (saliency 100))
```

```
  (MAIN::object (is-a Vepole) (:NAME ?nom))
```

=>

```
(unmake-instance ?nom))
```

```
(defrule RESOLUTION::standard_1.1.1a
```

```
  ?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (elements $?a
?substance&:(element-est-un Substance ?substance) $?b) (ajout_champ TRUE) (ajout_substance
TRUE) (completude FALSE))
```

```
  (not (MAIN::object (is-a Vepole) (:NAME ?n) (elements $?_2_c ?_2_s1&:(element-est-
un Substance ?_2_s1) ?_2_s2&:(element-est-un Substance ?_2_s2) $?_2_d)))
```

=>

```
(printout t "Completez " ?vepole " en ajoutant une substance et un champ d'interaction a
" ?substance "." crlf))
```

```
(defrule RESOLUTION::standard_1.1.1b
```

```
  ?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (elements $?a
?substance1&:(element-est-un Substance ?substance1) ?substance2&:(element-est-un Substance
?substance2) $?b) (ajout_champ TRUE) (completude FALSE))
```

=>

```
(printout t "Completez " ?vepole " en ajoutant un champ d'interaction a " ?substance1 " et
" ?substance2 "." crlf))
```

```
(defrule RESOLUTION::standard_1.1.1c
```

```
?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (elements $?a
?substance1&:(element-est-un Substance ?substance1) ?champ&:(element-est-un Substance
?champ) $?b) (ajout_substance TRUE) (completude FALSE))
```

=>

```
(printout t "Completez " ?vepole " en ajoutant une substance qui reagisse a " ?champ "
sous l'action de " ?substance1 "." crlf))
```

```
(defrule RESOLUTION::standard_1.1.2
```

```
?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (fonctionnalite FALSE) (elements
$a ?substance&:(element-est-un Substance ?substance) $b) (ajout_substance TRUE)
(completude TRUE))
```

```
(not (MAIN::object (is-a Vepole) (:NAME ?n) (elements $?_2_c ?_2_s1&:(element-est-
un Substance ?_2_s1) ?_2_s2&:(element-est-un Substance ?_2_s2) $?_2_d)))
```

=>

```
(printout t "Completez " ?vepole " en ajoutant une substance et un champ d'interaction a
" ?substance "." crlf))
```

```
(defrule RESOLUTION::standard_1.1.3
```

```
?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (fonctionnalite FALSE) (elements
$_2_c ?_2_s1&:(element-est-un Substance ?_2_s1) ?_2_s2&:(element-est-un Substance
?_2_s2) $_2_d) (ajout_additif FALSE) (completude TRUE))
```

=>

```
(printout t "Completez " ?vepole " en ajoutant une substance etrangere a " ?s1 " et " ?s2
", afin d'augmenter l'interaction entre les deux." crlf))
```

```
(defrule RESOLUTION::standard_1.1.4
```

```
?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (fonctionnalite FALSE) (elements
$_2_c ?_2_s1&:(element-est-un Substance ?_2_s1) ?_2_s2&:(element-est-un Substance
?_2_s2) $_2_d) (ajout_substance FALSE) (completude TRUE))
```

=>

(printout t "Completez " ?vepole " en ajoutant une substance issue de l'environnement a "
?s1 " et " ?s2 ", afin d'augmenter l'interaction entre les deux." crlf))

(defrule RESOLUTION::standard_1.1.5

?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (fonctionnalite FALSE) (elements
\$_2_c ?_2_s1&:(element-est-un Substance ?_2_s1) ?_2_s2&:(element-est-un Substance
?_2_s2) \$_2_d) (ajout_additif FALSE) (completude TRUE))

=>

(printout t "Completez " ?vepole " en ajoutant une substance etrangere a " ?s1 " et " ?s2
", afin d'augmenter l'interaction entre les deux." crlf))

(defrule RESOLUTION::standard_1.1.6

?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (fonctionnalite FALSE)
(completude TRUE) (interactions \$?a ?interaction&:(element-est-un Interaction ?interaction)
\$?b))

?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (fonctionnalite FALSE)
(completude TRUE) (interactions \$?a ?interaction&:(element-est-un Interaction ?interaction)
\$?b))

(MAIN::object (is-a Interaction) (:NAME ?name&:(eq (str-cat ?name) (str-cat
?interaction))) (caractere Excessive))

=>

(printout t "Enlevez l'exc?s de Substance par le champ et l'exc?s de champ par la
substance" crlf))

(defrule RESOLUTION::standard_1.2.1

?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (elements \$_2_e
?_2_s1&:(element-est-un Substance ?_2_s1) ?_2_s2&:(element-est-un Substance ?_2_s2)
\$_2_f) (necessite_de_contact FALSE) (interactions \$?a ?utile&:(element-est-un Interaction
?utile) \$?b))

```
(MAIN::object (is-a Vepole) (:NAME ?n) (interactions $?c ?nefaste&:(element-est-un
Interaction ?nefaste) $?d))
```

```
(MAIN::object (is-a Interaction) (:NAME ?name&:(eq (str-cat ?name) (str-cat ?utile)))
(caractere ?caractere&:(not (eq ?caractere Nefaste))))
```

```
(MAIN::object (is-a Interaction) (:NAME ?name&:(eq (str-cat ?name) (str-cat ?nefaste)))
(caractere Nefaste))
```

=>

```
(printout t "Eliminer " ?nefaste " en introduisant une substance etrangere entre " ?s1 " et "
?s2 "." crlf))
```

```
(defrule RESOLUTION::standard_1.2.2
```

```
?vepole <- (MAIN::object (is-a Vepole) (:NAME ?n) (elements $?_2_e
?_2_s1&:(element-est-un Substance ?_2_s1) ?_2_s2&:(element-est-un Substance ?_2_s2)
$?_2_f) (ajout_substance FALSE) (necessite_de_contact FALSE) (interactions $?a
?utile&:(element-est-un Interaction ?utile) $?b))
```

```
(MAIN::object (is-a Vepole) (:NAME ?n) (interactions $?c ?nefaste&:(element-est-un
Interaction ?nefaste) $?d))
```

```
(MAIN::object (is-a Interaction) (:NAME ?name&:(eq (str-cat ?name) (str-cat ?utile)))
(caractere ?caractere&:(not (eq ?caractere Nefaste))))
```

```
(MAIN::object (is-a Interaction) (:NAME ?name&:(eq (str-cat ?name) (str-cat ?nefaste)))
(caractere Nefaste))
```

=>

```
(printout t "Eliminer " ?nefaste " en introduisant une substance qui est une modification
de " ?s1 " et/ou " ?s2 "." crlf))
```

```
(defrule RESOLUTION::standard_1.2.3
```

```
(MAIN::object (is-a Vepole) (:NAME ?n) (interactions $?c ?nefaste&:(element-est-un
Interaction ?nefaste) $?d))
```

```
(MAIN::object (is-a Interaction) (:NAME ?name&:(eq (str-cat ?name) (str-cat ?nefaste)))
(caractere Nefaste))
```

=>

(printout t "Eliminer " ?nefaste " en introduisant un element qui attire " ?nefaste " sur
lui." crlf))

(defrule RETOUR::reset_pose

(MAIN::object (is-a Pose) (:NAME ?nom))

=>

(unmake-instance ?nom))

ANNEXE-VIII. LISTE DES QUESTIONS

; Tue May 04 17:04:26 CEST 2004

;

;+ (version "2.0.1")

;+ (build "Build 168")

([04] of Question

(type_reponse reponsemultiple)

(texte "Donnez brievement un liste des differents elements composant reponse02 ?")

(numero 4)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Ressources)

(action modifie))

([17] of Question

(type_reponse simple)

(texte "Quelle est la fonction principale que le futur systeme aura a realiser ?")

(numero 17)

(valide_incremente FALSE)

(slot [%3ANAME])

(classe Fonction)

(action modifie))

([18] of Question

(type_reponse multiferme)

(valide

"Oui"

"Non")

(texte "Existe-t-il un systeme realisant actuellement cette fonction ?")

(numero 18))

([19] of Question

(type_reponse simple)

(texte "Pourquoi le systeme existant, reponse22, realisant reponse17 ne vous satisfait-il pas ?")

(numero 19)

(slot [%3ANAME])

(classe Fonction)

(action modifie))

([22] of Question

(type_reponse multiouvert)

(texte "Quel est le nom de ce systeme ?")

(numero 22)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Systeme)

(action modifie))

([32] of Question

(type_reponse aucune)

(texte "Il est donc necessaire de: concevoir un systeme agissant sur reponse28, afin d'en faire passer le parametre reponse29 de reponse30 a reponse31.")

(numero 32)

(action rien))

([question00] of Question

(type_reponse simple)

(texte "Donnez un nom au projet.")

(numero 0)

(valide_incremente FALSE)

(slot [%3ANAME])

(classe Etude)

(action modifie))

([question01] of Question

(type_reponse multiouvert)

(texte "Quel est le nom du systeme a ameliorer ?")

(numero 1)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Systeme)

(action modifie))

([question02] of Question

(type_reponse multiferme)

(valide

"Oui"

"Non")

(texte "Cette fonction est-elle decomposable ?")

(numero 2)

(valide_incremente FALSE))

([question05] of Question

(type_reponse aucune)

(texte "Il faut donc resoudre la contradiction suivante: #contradiction\$parametre# de #fonction_utile\$produit# doit etre #contradiction\$valeur1# afin d'avoir #fonction_utile\$parametre# de #fonction_utile\$produit# a #fonction_utile\$valeur_objectif#. Mais #contradiction\$parametre# de #fonction_utile\$produit# ne doit pas etre #contradiction\$valeur1# pour eviter l'apparition de #fonction_nefaste\$:NAME#.")

(numero 5)

(valide_incremente FALSE)

(action rien))

([question06] of Question

(type_reponse aucune)

(texte "Le probleme est le suivant: il faut ameliorer la realisation de #fonction_utile\$:NAME#, mais vous ne savez pas comment faire. Utilisez les lois d'evolution des systemes techniques avec le systeme identifie.")

(numero 6)

(valide_incremente FALSE)

(action rien))

([question07] of Question

(type_reponse aucune)

(texte "Le probleme est le suivant: il faut assurer la realisation de #fonction_utile\$:NAME#, mais vous ne savez pas comment faire. Il est necessaire d'identifier un element pouvant agir sur #fonction_utile\$produit# afin de faire passer #fonction_utile\$parametre# a #fonction_utile\$valeur_finale#.")

(numero 7)

(valide_incremente FALSE)

(action rien))

([question15] of Question

(type_reponse multiferme)

(valide "Amelioration d'un systeme existant.")

(texte "Quel est l'objectif de l'etude ?")

(numero 15)

(valide_incremente FALSE)

(slot [objectif])

(objet "reponse00")

(classe Etude)

(action modifie))

([question16] of Question

(type_reponse multiferme)

(valide

 "Amelioration d'une fonction utile."

 "Ajout d'une nouvelle fonctionnalite."

 "Supression d'un effet non desire.")

(texte "De quelle maniere souhaitez-vous ameliorer le systeme reponse01 ?")

(numero 16)

(valide_incremente FALSE)

(slot [objectif])

(objet "reponse00")

(classe Etude)

(action modifie))

([question20] of Question

(type_reponse multiouvert)

(texte "Quelle est la fonction dont vous souhaitez ameliorer la realisation ?")

(numero 20)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Fonction)

(action modifie))

([question21] of Question

(type_reponse multiouvert)

(texte "Quel est l'effet genant, la fonction genante, a supprimer ? (formulation conseillee:
un verbe + un COD)")

(numero 21)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Fonction)

(action modifie))

([question23] of Question

(type_reponse multiouvert)

(texte "Definissez une sous-fonction de reponse20.")

(numero 23)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Fonction)

(action modifie))

([question24] of Question

(type_reponse multiferme)

(valide

"Oui"

"Non")

(texte "Y'a-t-il d'autres sous-fonctions de reponse20?")

(numero 24))

([question25] of Question

(type_reponse multiferme)

(texte "Parmi les fonctions suivantes, quelle est celle dont la realisation pose le plus de problemes ?")

(numero 25)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Fonction))

([question26] of Question

(type_reponse simple)

(texte "Il faut ameliorer la realisation de #fonction_utilite\$:NAME#. Quelle serait la valeur ideale du parametre #fonction_utilite\$parametre# de #fonction_utilite\$produit# apres la realisation de #fonction_utilite\$:NAME#?")

(numero 26)

(valide_incremente FALSE)

(slot [valeur_objectif])

(objet "reponse25")

(action modifie))

([question27] of Question

(type_reponse multiouvert)

(texte "Quelle est la fonction que vous souhaitez ajouter au systeme reponse01 ?")

(numero 27)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Fonction)

(action modifie))

([question28] of Question

(type_reponse multiouvert)

(texte "Quel parametre de #fonction_utile\$produit# faut-il modifier afin d'avoir #fonction_utile\$parametre# a la valeur #fonction_utile\$valeur_objectif# ?")

(numero 28)

(valide_incremente TRUE)

(slot [parametre])

(objet "contradiction")

(action modifie))

([question29] of Question

(type_reponse simple)

(texte "Quelle est la valeur #contradiction\$parametre# de #fonction_utile\$produit# qui permet de mettre #fonction_utile\$parametre# a #fonction_utile\$valeur_objectif# ?")

(numero 29)

(valide_incremente FALSE)

(slot [valeur1])

(objet "contradiction")

(action modifie))

([question30] of Question

(type_reponse multiferme)

(valide

"Oui"

"Non")

(texte "Si #contradiction\$parametre# de #fonction_utile\$produit# vaut #contradiction\$valeur1#. Cela cree-t-il un effet genant ?")

(numero 30))

([question31] of Question

(type_reponse multiouvert)

(texte "Si on considere que l'effet genant est la realisation d'une fonction non souhaitee. Quelle est cette fonction ?")

(numero 31)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Fonction)

(action modifie))

([question37] of Question

(type_reponse aucune)

(texte "Il faut modifier le systeme pour agir sur #fonction_utile\$produit#, afin de faire passer #fonction_utile\$parametre# de #fonction_utile\$valeur_initiale# a #fonction_utile\$valeur_finale#.")

(numero 37)

(valide_incremente FALSE)

(action rien))

([question38] of Question

(type_reponse multiferme)

(valide

"Oui"

"Non")

(texte "Si #fonction_utile\$parametre# de #fonction_utile\$produit# vaut #fonction_utile\$valeur_finale#. Cela cree-t-il un effet genant, ailleurs dans le systeme ?")

(numero 38)

(valide_incremente FALSE))

([question39] of Question

(type_reponse multiouvert)

(texte "Si on considere que l'effet genant est la realisation d'une fonction non souhaitee. Quelle est cette fonction ?")

(numero 39)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Fonction)

(action modifie))

([question40] of Question

(type_reponse aucune)

(texte "Nous avons donc la contradiction suivante: #fonction_utile\$parametre# doit etre #fonction_utile\$valeur_finale# afin d'assurer la realisation de #fonction_utile\$:NAME#, mais

#fonction_utile\$parametre# ne doit pas etre #fonction_utile\$valeur_finale# afin d'eviter l'apparition de #fonction_nefaste\$:NAME#.")

(numero 40)

(valide_incremente FALSE)

(action rien))

([question43] of Question

(type_reponse multiouvert)

(texte "Quel parametre de #systeme_nefaste\$outil# peut-on modifier afin d'empecher l'apparition #fonction_nefaste\$:NAME# ?")

(numero 43)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Parametre)

(action modifie))

([question44] of Question

(type_reponse simple)

(texte "Quelle valeur #systeme_nefaste\$parametre# de #systeme_nefaste\$outil# permet d'empecher l'apparition de #fonction_nefaste\$:NAME# ?")

(numero 44)

(valide_incremente FALSE)

(slot [valeur_objectif])

(objet "reponse21")

(action modifie))

([question45] of Question

(type_reponse multiferme)

(valide

"Oui"

"Non")

(texte "Si #fonction_nefaste\$parametre# de #systeme_nefaste\$outil# est a #fonction_nefaste\$valeur_objectif#, le probleme est-il resolu ?")

(numero 45)

(valide_incremente FALSE))

([question46] of Question

(type_reponse simple)

(texte "Quelle est la valeur actuelle pour #fonction_nefaste\$parametre# de #systeme_nefaste\$outil# ?")

(numero 46)

(valide_incremente FALSE)

(slot [valeur2])

(objet "contradiction_outil")

(action modifie))

([question47] of Question

(type_reponse multiouvert)

(texte "Si nous ne pouvons le changer, c'est que le fait d'avoir #fonction_nefaste\$parametre# de #systeme_nefaste\$outil# a #contradiction_outil\$valeur2# permet de realiser une action utile. Quelle est cette fonction ?")

(numero 47)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Fonction)

(action modifie))

([question48] of Question

(type_reponse aucune)

(texte "La contradiction suivante est alors a resoudre: #fonction_nefaste\$parametre# de #systeme_nefaste\$outil# doit etre #contradiction_outil\$valeur2# pour assurer la realisation de #contradiction_outil\$fonction2#, mais #fonction_nefaste\$parametre# de #systeme_nefaste\$outil# doit etre #contradiction_outil\$valeur1# pour #contradiction_outil\$fonction1#.")

(numero 48)

(valide_incremente FALSE)

(action rien))

([question50] of Question

(type_reponse multiouvert)

(texte "Nous allons analyser le systeme responsable de l'apparition de #fonction_nefaste\$:NAME#. Donnez un nom a ce systeme.")

(numero 50)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Systeme)

(action modifie))

([question52] of Question

(type_reponse multiouvert)

(texte "Quelle est la fonction utile qui n'est pas assuree si #fonction_nefaste\$parametre# n'est pas #fonction_nefaste\$valeur_initiale# ?")

(numero 52)

(valide_incremente TRUE)

(slot [%3ANAME])

(classe Fonction)

(action modifie))

([question53] of Question

(type_reponse aucune)

(texte "Il faut donc resoudre la contradiction suivante: #fonction_nefaste\$parametre# doit etre #fonction_nefaste\$valeur_finale# afin d'assurer la realisation de #fonction_utile\$:NAME#, mais #fonction_nefaste\$parametre# ne doit pas etre #fonction_nefaste\$valeur_finale# afin d'eviter l'apparition de #fonction_nefaste\$:NAME#.")

(numero 53)

(valide_incremente FALSE)

(action rien))

([question54] of Question

(type_reponse multiferme)

(valide

"cela PARTICIPE a la realisation d'un fonction utile."

"cela est la CONSEQUENCE de la realisation d'une fonction utile")

(texte "Quelle est la cause de #fonction_nefaste\$parametre# #fonction_nefaste\$valeur_finale# ?")

(numero 54)

(valide_incremente FALSE))

([question55] of Question

(type_reponse multiouvert)

(texte "Quelle est la fonction utile qui n'est pas assuree si #fonction_nefaste\$parametre# n'est pas #fonction_nefaste\$valeur_finale# ?")

(numero 55)

(valide_incremente FALSE)

(slot [%3ANAME])

(classe Fonction)

(action modifie))

([question56] of Question

(type_reponse aucune)

(texte "Il faut donc resoudre la contradiction suivante: #fonction_nefaste\$parametre# est #fonction_nefaste\$valeur_finale# car la realisation de #fonction_utile\$:NAME# est necessaire, mais #fonction_nefaste\$parametre# ne doit pas etre #fonction_nefaste\$valeur_finale# afin d'eviter l'apparition de #fonction_nefaste\$:NAME#.")

(numero 56)

(valide_incremente FALSE)

(action rien))

ANNEXE-IX. LISTE DES FONCTIONS DEFINIES DANS JESS

```
(deffunction affiche_aucune ()
  (bind ?*fenetrequestion* (new JFrame "Commentaire"))
  (bind ?*panneureponse* (new JPanel))
  (bind ?*question* (new JTextArea))
  (bind ?*barrequestion* (new JScrollPane ?*question*))
  (bind ?*reponseoksimple* (new JButton OK))
  (bind ?*boutonexit* (new JButton QUITTE))
  (call ?*fenetrequestion* setSize 500 200)
  (call ?*question* setSize 20 100)
  (call ?*reponseoksimple* addActionListener (new jess.awt.ActionListener
suivant_aucune (engine)))
  (call ?*panneureponse* add ?*reponseoksimple*)
  (call ?*boutonexit* addActionListener (new jess.awt.ActionListener sortir (engine)))
  (call ?*panneureponse* add ?*boutonexit*)
  (call ?*fenetrequestion* validate)
  (call ?*question* setText (str-cat ?question))
  (call ?*question* setEditable FALSE)
  (call (call ?*fenetrequestion* getContentPane) add ?*barrequestion*)
  (call (call ?*fenetrequestion* getContentPane) add ?*panneureponse* (get-member
BorderLayout SOUTH))
  (call ?*fenetrequestion* validate)
  (call ?*fenetrequestion* setVisible TRUE))
```

```

(deffunction affiche_fonction (?fonction ?produit ?parametre ?valeur_initiale
?valeur_finale)
  (bind ?*fenetrequestion* (new JFrame "Veuillez repondre"))
  (bind ?*panneaufonction* (new JPanel))
  (bind ?*question* (new JTextArea))
  (bind ?*barrequestion* (new JScrollPane ?*question*))
  (bind ?*boutonexit* (new JButton FERME))
  (call ?*question* setSize 20 100)
  (call (call ?*fenetrequestion* getContentPane) add ?*barrequestion*)
  (call ?*boutonexit* addActionListener (new jess.awt.ActionListener sortir (engine)))
  (call ?*panneaufonction* add ?*boutonexit*)
  (call (call ?*fenetrequestion* getContentPane) add ?*panneaufonction* (get-member
BorderLayout SOUTH))
  (if (eq ?parametre nil) then
    (bind ?parametre ""))
  (if (eq ?valeur_initiale nil) then
    (bind ?valeur_initiale ""))
  (if (eq ?valeur_finale nil) then
    (bind ?valeur_finale ""))
  (call ?*question* setText (str-cat (create$ "La fonction : " ?fonction crlf "agit sur : "
?produit crlf "dont elle modifie : " ?parametre crlf "en le faisant passer de : " ?valeur_initiale crlf
"? : " ?valeur_finale)))
  (call ?*fenetrequestion* pack)
  (call ?*fenetrequestion* validate)
  (call ?*question* setEditable FALSE)
  (call ?*fenetrequestion* setVisible TRUE))

(deffunction affiche_menu ()

```

```
(bind ?*fenetremenu* (new JFrame "Menu"))

(bind ?*bouton_nouvelle_etude* (new JButton "NOUVELLE ETUDE"))

(call ?*bouton_nouvelle_etude* addActionListener (new jess.awt.ActionListener
nouvelle_etude (engine)))

(call (call ?*fenetremenu* getContentPane) add ?*bouton_nouvelle_etude* (get-member
BorderLayout NORTH))

(bind ?*bouton_reprendre_etude* (new JButton "REPRENDRE ETUDE"))

(call ?*bouton_reprendre_etude* addActionListener (new jess.awt.ActionListener
reprendre_etude (engine)))

(call (call ?*fenetremenu* getContentPane) add ?*bouton_reprendre_etude* (get-
member BorderLayout CENTER))

(bind ?*bouton_initialiser_domaine* (new JButton "INITIALISER DOMAINE"))

(call ?*bouton_initialiser_domaine* addActionListener (new jess.awt.ActionListener
initialiser_domaine (engine)))

(call (call ?*fenetremenu* getContentPane) add ?*bouton_initialiser_domaine* (get-
member BorderLayout SOUTH))

(call ?*fenetremenu* pack)

(call ?*fenetremenu* validate)

(call ?*fenetremenu* toFront)

(call ?*fenetremenu* setVisible TRUE)

(return))

(deffunction affiche_multiferme (?question ?valide)

(bind ?*fenetrequestion* (new JFrame "Veuillez repondre"))

(bind ?*panneureponse* (new JPanel))

(bind ?*question* (new JTextArea))

(bind ?*barrequestion* (new JScrollPane ?*question*))

(bind ?*liste* (new JComboBox))
```

```
(bind ?*retour* (new JButton RETOUR))
(bind ?*reponseokferme* (new JButton OK))
(bind ?*boutonexit* (new JButton QUITTE))
(bind ?*reponse* (new JTextField 20))
(call ?*question* setSize 20 100)
(call ?*panneureponse* add ?*reponse*)
(call ?*reponse* setText ?*repondu*)
(call ?*reponse* setEditable FALSE)
(call (call ?*fenetrequestion* getContentPane) add ?*barrequestion*)
(call ?*retour* addActionListener (new jess.awt.ActionListener retour (engine)))
(call ?*panneureponse* add ?*retour*)
(call ?*liste* addActionListener (new jess.awt.ActionListener choix (engine)))
(call ?*reponseokferme* addActionListener (new jess.awt.ActionListener
lire_reponse_simple (engine)))
(call ?*panneureponse* add ?*reponseokferme*)
(call ?*boutonexit* addActionListener (new jess.awt.ActionListener sortir (engine)))
(call ?*panneureponse* add ?*boutonexit*)
(call ?*panneureponse* add ?*liste*)
(call (call ?*fenetrequestion* getContentPane) add ?*panneureponse* (get-member
BorderLayout SOUTH))
(call ?*liste* removeAllItems)
(foreach ?item ?valide (call ?*liste* addItem ?item))
(call ?*question* setText (str-cat ?question))
(call ?*question* setEditable FALSE)
(call ?*fenetrequestion* pack)
(call ?*fenetrequestion* validate)
(call ?*fenetrequestion* setVisible TRUE))
```

```
(deffunction affiche_multiouvert (?question ?valide)
  (bind ?*fenetrequestion* (new JFrame "Veuillez repondre"))
  (bind ?*panneureponse* (new JPanel))
  (bind ?*question* (new JTextArea))
  (bind ?*barrequestion* (new JScrollPane ?*question*))
  (bind ?*liste* (new JComboBox))
  (call ?*liste* setEditable FALSE)
  (bind ?*reponse* (new JTextField 20))
  (bind ?*retour* (new JButton RETOUR))
  (bind ?*reponseokferme* (new JButton OK))
  (bind ?*boutonexit* (new JButton QUITTE))
  (call ?*question* setSize 20 100)
  (call ?*panneureponse* add ?*reponse*)
  (call ?*reponse* setText ?*repondu*)
  (call ?*liste* addActionListener (new jess.awt.ActionListener choix (engine)))
  (call (call ?*fenetrequestion* getContentPane) add ?*barrequestion*)
  (call ?*retour* addActionListener (new jess.awt.ActionListener retour (engine)))
  (call ?*panneureponse* add ?*retour*)
  (call ?*reponseokferme* addActionListener (new jess.awt.ActionListener
lire_reponse_simple (engine)))
  (call ?*panneureponse* add ?*reponseokferme*)
  (call ?*boutonexit* addActionListener (new jess.awt.ActionListener sortir (engine)))
  (call ?*panneureponse* add ?*boutonexit*)
  (call ?*panneureponse* add ?*liste*)
  (call (call ?*fenetrequestion* getContentPane) add ?*panneureponse* (get-member
BorderLayout SOUTH))
  (call ?*liste* removeAllItems)
  (foreach ?item ?valide (call ?*liste* addItem ?item))
```

```
(call ?*question* setText (str-cat ?question))
(call ?*question* setEditable FALSE)
(call ?*fenetrequestion* pack)
(call ?*fenetrequestion* validate)
(call ?*fenetrequestion* setVisible TRUE))

(deffunction affiche_reponsemultiple (?question ?valide)
  (bind ?*fenetrequestion* (new JFrame "Veuillez repondre"))
  (bind ?*panneareponse* (new JPanel))
  (bind ?*question* (new JTextArea))
  (bind ?*barrequestion* (new JScrollPane ?*question*))
  (bind ?*listemodele* (new DefaultListModel))
  (bind ?*listemultiple* (new JList ?*listemodele*))
  (call ?*listemultiple* setSelectionMode (get-member ListSelectionMode
MULTIPLE_INTERVAL_SELECTION))
  (call ?*listemultiple* setLayoutOrientation (get-member JList HORIZONTAL_WRAP))
  (call ?*listemultiple* setVisibleRowCount 1)
  (call ?*listemultiple* setSelectedIndex 0)
  (bind ?*scroll* (new JScrollPane ?*listemultiple*))
  (call ?*scroll* setPreferredSize (new java.awt.Dimension 250 80))
  (bind ?*retour* (new JButton RETOUR))
  (bind ?*retirer* (new JButton RETIRER))
  (call ?*retour* addActionListener (new jess.awt.ActionListener retour (engine)))
  (call ?*panneareponse* add ?*retour*)
  (call ?*retirer* addActionListener (new jess.awt.ActionListener retirer_element
(engine)))
  (call ?*retirer* setEnabled TRUE)
  (bind ?*ajout_element* (new JTextField 20))
```

```
(bind ?*ajouter* (new JButton AJOUTER))
(call ?*ajouter* addActionListener (new jess.awt.ActionListener ajouter_element
(engine)))
(bind ?*reponseokferme* (new JButton OK))
(bind ?*boutonexit* (new JButton QUITTE))
(call ?*question* setSize 20 100)
(call (call ?*fenetrequestion* getContentPane) add ?*barrequestion* (get-member
BorderLayout NORTH))
(call ?*reponseokferme* addActionListener (new jess.awt.ActionListener
lire_reponsemultiple (engine)))
(call ?*panneureponse* add ?*listemultiple*)
(call ?*panneureponse* add ?*retirer*)
(call ?*panneureponse* add ?*ajouter*)
(call ?*panneureponse* add ?*ajout_element*)
(call ?*panneureponse* add ?*reponseokferme*)
(call ?*boutonexit* addActionListener (new jess.awt.ActionListener sortir (engine)))
(call ?*panneureponse* add ?*boutonexit*)
(call (call ?*fenetrequestion* getContentPane) add ?*listemultiple* (get-member
BorderLayout CENTER))
(call (call ?*fenetrequestion* getContentPane) add ?*panneureponse* (get-member
BorderLayout SOUTH))
(foreach ?item ?valide (call ?*listemultiple* addItem ?item))
(call ?*question* setText (str-cat ?question))
(call ?*question* setEditable FALSE)
(call ?*fenetrequestion* pack)
(call ?*fenetrequestion* validate)
(call ?*fenetrequestion* setVisible TRUE))
```

```
(deffunction affiche_simple (?question)
  (bind ?*fenetrequestion* (new JFrame "Veuillez repondre"))
  (bind ?*panneureponse* (new JPanel))
  (bind ?*question* (new JTextArea))
  (bind ?*barrequestion* (new JScrollPane ?*question*))
  (bind ?*reponse* (new JTextField 20))
  (bind ?*retour* (new JButton RETOUR))
  (bind ?*reponseoksimple* (new JButton OK))
  (bind ?*boutonexit* (new JButton QUITTE))
  (call ?*question* setSize 20 100)
  (call ?*panneureponse* add ?*reponse*)
  (call ?*retour* addActionListener (new jess.awt.ActionListener retour (engine)))
  (call ?*panneureponse* add ?*retour*)
  (call ?*reponseoksimple* addActionListener (new jess.awt.ActionListener
lire_reponse_simple (engine)))
  (call ?*panneureponse* add ?*reponseoksimple*)
  (call ?*boutonexit* addActionListener (new jess.awt.ActionListener sortir (engine)))
  (call ?*panneureponse* add ?*boutonexit*)
  (call ?*reponse* setText ?*repondu*)
  (call ?*fenetrequestion* validate)
  (call ?*question* setText (str-cat ?question))
  (call ?*question* setEditable FALSE)
  (call (call ?*fenetrequestion* getContentPane) add ?*barrequestion*)
  (call (call ?*fenetrequestion* getContentPane) add ?*panneureponse* (get-member
BorderLayout SOUTH))
  (call ?*fenetrequestion* pack)
  (call ?*fenetrequestion* validate)
  (call ?*fenetrequestion* setVisible TRUE))
```

```
(deffunction ajoute_parametre (?objet ?parametre ?valeur)
  (bind ?slot (slot-get ?objet ?parametre))
  (bind ?test FALSE)
  (foreach ?item ?slot (if (eq (instance-name ?item) ?valeur) then (bind ?test TRUE)))
  (if (eq ?test FALSE) then
    (printout t "le slot n'existe pas" crlf)
    (if (eq (str-cat (slot-types (class ?objet) ?parametre)) "CLS INSTANCE") then
      (printout t "class INSTANCE" crlf)
      (if (not (instancep ?valeur)) then
        (printout t "la valeur n'existe pas" crlf)
        (if (> (length$ (slot-allowed-classes (class ?objet) ?parametre)) 1) then
          (printout t "plus d'une classe" crlf)
          (bind ?class "")
          (while (eq ?class "") do
            (bind ?class (str-cat (pose_question (str-cat ?valeur " est : ") multifерme
(slot-allowed-classes (class ?objet) ?parametre))))))
          else
            (printout t "une seule classe" crlf)
            (bind ?class (str-cat (slot-allowed-classes (class ?objet) ?parametre))))
            (make-instance ?valeur of ?class)))
        (if (> (length$ (slot-cardinality (class ?objet) ?parametre)) 1) then
          (printout t "cardinalite >1" crlf)
          (slot-insert$ ?objet ?parametre 1 ?valeur)
        else
          (printout t "cardinalite = 1" crlf)
          (slot-set ?objet ?parametre ?valeur)))
```

```
(reset))
```

```
(deffunction ajouter_element (?event)
```

```
  (bind ?nom (call ?*ajout_element* getText))
```

```
  (if (eq ?nom "") then
```

```
    (call ?*ajout_element* requestFocusInWindow)
```

```
    (call ?*ajout_element* selectAll)
```

```
  (return))
```

```
(bind ?index (call ?*listemultiple* getSelectedIndex))
```

```
(printout t ?index)
```

```
(printout t crlf ?nom crlf ?index crlf)
```

```
(call ?*listemodele* insertElementAt ?nom ?index)
```

```
(call ?*ajout_element* requestFocusInWindow)
```

```
(call ?*ajout_element* setText "")
```

```
(call ?*listemultiple* setSelectedIndex ?index)
```

```
(call ?*listemultiple* ensureIndexIsVisible ?index))
```

```
(deffunction choix ()
```

```
  (call ?*reponse* setText (call ?*liste* getSelectedItem)))
```

```
(deffunction cree_multiple (?objet ?slot)
```

```
  (foreach ?instance of ?*repondu* (make-instance ?instance of (str-cat ?objet)) (slot-set  
?instance ?slot ?instance)))
```

```
(deffunction cree_reponse (?num ?type)
```

```
  (if (eq (str-length ?*index*) 1) then
```

```
    (bind ?reponse (str-cat reponse0 ?*index*)))
```

```
else
  (bind ?reponse (str-cat reponse ?*index*))
(if (eq (str-length ?num) 1) then
  (bind ?reutilise (str-cat reponse0 ?num))
else
  (bind ?reutilise (str-cat reponse ?num)))
(make-instance ?reponse of Reponse)
(bind ?*index* (+ ?*index* 1))
(bind ?*recherche* ?*index*)
(slot-set ?reponse index ?*index*)
(slot-set ?reponse numero ?num)
(slot-set ?reponse reutilisation ?reutilise)
(if (eq ?type "reponsemultiple") then
  (slot-set ?reponse reponse_multiple ?*repondu*)
else
  (slot-set ?reponse reponse ?*repondu*))
(bind ?*repondu* ""))

(defun demande_simple (?question ?valide)
  (bind ?*valide* ?valide)
  (affiche ?question lire_reponse_simple)
  (bind ?*repondu* "")
  (while (eq ?*repondu* "") do)
  (return ?*repondu*))

(defun element-est-un (?type ?element)
  (bind ?nature (slot-get ?element :DIRECT-TYPE))
```

```
(if (eq (str-compare ?nature ?type) 0) then
  TRUE
else
  FALSE))
```

```
(deffunction fermefenetre ()
  (call ?*fenetrequestion* setVisible FALSE)
  (call ?*fenetrequestion* dispose))
```

```
(deffunction initialise_etudes ()
  (clear-focus-stack)
  (focus RESET_ETUDE)
  (run)
  (clear-focus-stack)
  (make-instance 0 of Pose)
  (slot-set "0" numero 0)
  (return))
```

```
(deffunction initialise_index ()
  (focus INITIALISE)
  (run)
  (return))
```

```
(deffunction initialiser_domaine ()
  (call ?*fenetremenu* setVisible FALSE)
  (clear-focus-stack)
  (focus RESET_DOMAINE))
```

```
(run)
(affiche_menu)
(return))
```

```
(deffunction lire_reponse_simple (?event)
  (printout t (call ?*reponse* getText))
  (bind ?*repondu* (call ?*reponse* getText))
  (printout t ?*repondu* crlf)
  (call ?*fenetrequestion* setVisible FALSE)
  (return (call ?*reponse* getText)))
```

```
(deffunction lire_reponsemultiple (?event)
  (bind ?*repondu* (call ?*listemultiple* getSelectedValues))
  (printout t crlf ?*repondu* crlf)
  (return ?*repondu*))
```

```
(deffunction liste_valides (?type)
  (bind ?valides (create$))
  (bind ?type (sym-cat ?type))
  (bind ?liste (run-query MAIN::liste_valides ?type))
  (while (call ?liste hasNext)
    (bind ?next (call ?liste next))
    (bind ?valide (call ?next fact 1))
    (bind ?nom (fact-slot-value ?valide :NAME))
    (bind ?valides (insert$ ?valides 1 ?nom)))
  (return ?valides))
```

```
(deffunction modifie (?objet ?slot)
  (if (> (str-length ?objet) 7) then
    (if (eq (str-cat reponse) (sub-string 1 7 ?objet)) then
      (bind ?num (integer (sub-string 8 (str-length ?objet) ?objet)))
      (printout t ?num crlf)
      (assert (MAIN::modifie (num ?num) (slot ?slot)))
      (printout t crlf ?num ?slot (str-cat ?slot) crlf)
      (return)))
    (if (neq ?slot nil) then
      (if (and (eq (class ?objet) :STANDARD-CLASS) (not (instance-existp ?*repondu*)))
        then
          (make-instance ?*repondu* of ?objet))
        (slot-set ?*repondu* ?slot ?*repondu*)))

(deffunction nouvelle_etude ()
  (call ?*fenetremenu* setVisible FALSE)
  (initialise_etudes)
  (printout t ok)
  (reset-global)
  (clear-focus-stack)
  (focus MAIN)
  (run)
  (clear-focus-stack)
  (focus INTERFACE)
  (run)
  (bind ?*index* 0)
  (mapclass Pose)
```

(clear-focus-stack)

(focus QUESTION)

(run))

(deffunction pose_question (?question ?type ?valide)

(bind ?*valide* ?valide)

(bind ?*quitter* FALSE)

(if (eq (str-cat ?type) "simple") then

(affiche_simple ?question))

(if (eq (str-cat ?type) "aucune") then

(affiche_aucune ?question))

(if (eq (str-cat ?type) "multiferme") then

(affiche_multiferme ?question ?valide))

(if (eq (str-cat ?type) "reponsemultiple") then

(affiche_reponsemultiple ?question ?valide))

(if (eq (str-cat ?type) "multiouvert") then

(affiche_multiouvert ?question ?valide))

(bind ?*repondu* ""))

(while (eq ?*repondu* "") do

(if (eq ?*quitter* TRUE) then

(printout t "quitter" ?*quitter* crlf)

(return)))

(return ?*repondu*))

(deffunction question-precedente ()

(focus RETOUR)

(printout t ok4 crlf)

```
(run)
(printout t ok5 crlf)
(bind ?liste (run-query MAIN::question_precedente ?*recherche*))
(if (call ?liste hasNext) then
  (while (call ?liste hasNext)
    (bind ?next (call ?liste next))
    (bind ?reponse (call ?next fact 1))
    (bind ?listenumero (fact-slot-value ?reponse numero))
    (foreach ?item ?listenumero (bind ?numero ?item))
    (bind ?listereponse (fact-slot-value ?reponse reponse))
    (foreach ?item ?listereponse (bind ?*repondu* ?item)))
  else
    (bind ?numero 0)
    (bind ?*recherche* (- ?*recherche* 1))
    (make-instance (str-cat ?numero) of Pose)
    (slot-set (str-cat ?numero) numero (integer ?numero))
    (focus QUESTION)
    (run 1))
```

```
(deffunction reprendre_etude ()
  (reset)
  (initialise_index)
  (call ?*fenetremenu* setVisible FALSE)
  (clear-focus-stack)
  (focus QUESTION)
  (run))
```

```
(deffunction reset-global ()
```

```
  (bind ?*repondu* "")
```

```
  (bind ?*reponse* "")
```

```
  (bind ?*valide* nil))
```

```
(deffunction retirer_element (?event)
```

```
  (bind ?index (call ?*listemultiple* getSelectedIndex))
```

```
  (call ?*listemodele* remove ?index)
```

```
  (bind ?taille (call ?*listemultiple* getSize))
```

```
  (if (eq ?size 0) then
```

```
    (call ?*retirer setEnabled FALSE)
```

```
  else
```

```
    (if (eq ?index (call ?*listemultiple* getSize)) then
```

```
      (bind ?index (- ?index 1)))
```

```
    (call ?*listemultiple* setSelectedIndex ?index)
```

```
    (call ?*listemultiple* ensureIndexIsVisible ?index)))
```

```
(deffunction retour ()
```

```
  (bind ?*quitter* TRUE)
```

```
  (printout t ?*quitter* crlf)
```

```
  (fermefenetre)
```

```
  (bind ?*precedent* TRUE)
```

```
  (return))
```

```
(deffunction selectionner (?event)
```

```
  (if (not (call ?event getValueIsAdjusting)) then
```

```
    (if (eq (call ?*listemultiple* getSelectedIndex) -1) then
```

```
(call ?*retirer* setEnabled FALSE)
else
  (call ?*retirer* setEnabled TRUE))))

(deffunction sortir ()
  (printout t sortir crlf)
  (bind ?*quitter* TRUE)
  (fermefenetre)
  (return ?*quitter*))

(deffunction suivant_aucune (?event)
  (call ?*fenetrequestion* setVisible FALSE)
  (return "nil"))

(deffunction valide_texte (?texte)
  (bind ?question (create$))
  (while (or (neq FALSE (str-index # ?texte)) (neq FALSE (str-index reponse ?texte))) do
    (if (and (neq FALSE (str-index # ?texte)) (or (eq FALSE (str-index reponse ?texte))
    (and (neq FALSE (str-index reponse ?texte)) (< (str-index # ?texte) (str-index reponse ?texte))))))
  then
    (if (neq (str-index # ?texte) 1) then
      (bind ?question (insert$ ?question (+ 1 (length$ ?question)) (sub-string 1 (- (str-
index # ?texte) 1) ?texte)))
      (bind ?texte (sub-string (+ (str-index # ?texte) 1) (str-length ?texte) ?texte)))
      (bind ?objet (sub-string 1 (- (str-index $ ?texte) 1) ?texte))
      (bind ?slot (sub-string (+ (str-index $ ?texte) 1) (- (str-index # ?texte) 1) ?texte))
      (bind ?cherchenom (run-query MAIN::cherche_role ?objet))
      (bind ?ajout (create$))
```

```

(while (call ?cherchenom hasNext)
  (bind ?next (call ?cherchenom next))
  (bind ?token (call ?next fact 1))
  (bind ?liste_objet (fact-slot-value ?token :NAME))
  (foreach ?nom_objet ?liste_objet (if (eq (str-cat (slot-types (class ?nom_objet)
?slot)) "CLS INSTANCE") then (bind ?liste (slot-get ?nom_objet ?slot)) (bind ?cherchenomobjet
(run-query MAIN::cherche_objets ?liste)) (while (call ?cherchenomobjet hasNext) (bind
?nextobjet (call ?cherchenomobjet next)) (bind ?tokenobjet (call ?nextobjet fact 1)) (bind
?listeajout (fact-slot-value ?tokenobjet :NAME))) else (bind ?listeajout (fact-slot-value ?token
?slot))))))

  (bind ?ajout (insert$ ?ajout 1 ?listeajout)))

  (bind ?texte (sub-string (+ (str-index # ?texte) 1) (str-length ?texte) ?texte))

  (bind ?question (insert$ ?question (+ 1 (length$ ?question)) ?ajout)))

  (if (and (neq FALSE (str-index reponse ?texte)) (or (eq FALSE (str-index # ?texte))
(and (neq FALSE (str-index # ?texte)) (> (str-index # ?texte) (str-index reponse ?texte)))))) then

    (if (neq (str-index reponse ?texte) 1) then

      (bind ?question (insert$ ?question (+ 1 (length$ ?question)) (sub-string 1 (- (str-
index reponse ?texte) 1) ?texte)))

      (bind ?reponse (sub-string (str-index reponse ?texte) (+ (str-index reponse ?texte)
8) ?texte))

      (printout t ?reponse crlf)

      (bind ?liste (run-query MAIN::cherche_reponse (str-cat ?reponse)))

      (bind ?index 0)

      (while (call ?liste hasNext)

        (bind ?next (call ?liste next))

        (bind ?valide (call ?next fact 1))

        (bind ?test (fact-slot-value ?valide index))

        (foreach ?item ?test (if (> (integer ?item) ?index) then (bind ?index (integer
?item)) (bind ?reponse (fact-slot-value ?valide reponse))))))

```

```
(bind ?question (insert$ ?question (+ 1 (length$ ?question)) ?reponse))  
(bind ?texte (sub-string (+ (str-index reponse ?texte) 9) (str-length ?texte)  
?texte))  
else  
(if (eq (str-index reponse ?texte) 1) then  
  (bind ?reponse (sub-string 1 9 ?texte))  
  (bind ?liste (run-query MAIN::cherche_reponse ?reponse))  
  (bind ?index 0)  
  (while (call ?liste hasNext)  
    (bind ?next (call ?liste next))  
    (bind ?valide (call ?next fact 1))  
    (bind ?test (fact-slot-value ?valide index))  
    (foreach ?item ?test (if (> (integer ?item) ?index) then (bind ?index (integer  
?item)) (bind ?reponse (fact-slot-value ?valide reponse))))))  
  (bind ?question (insert$ ?question (+ 1 (length$ ?question)) ?reponse))  
  (bind ?texte (sub-string 10 (str-length ?texte) ?texte))))))  
(bind ?question (insert$ ?question (+ 1 (length$ ?question)) ?texte))  
(return ?question))
```

ANNEXE-X. PREMIERE SERIE D'EXPERIMENTATIONS

Donnez un nom au projet.

Projet TRIZ

Quel est l'objectif de l'etude ?

Amelioration d'un systeme existant.

Quel est le nom du systeme a ameliorer ?

Fer a souder

Quelle est la fonction principale réalisée par " Fer a souder " ?

Souder deux pieces

De quelle maniere souhaitez-vous ameliorer le systeme " Fer a souder " ?

Suppression d'un effet nefaste.

Quel est l'effet genant, la fonction genante, a supprimer ? (formulation conseillée: un verbe + un COD)

Bruler la main

La realisation d'une fonction s'effectue dans le but de modifier une caracteristique d'un objet. Quel est l'objet en question, pour la fonction " Bruler la main " ?

La main

" La main " est-il : (Substance, Systeme)

Substance

" Bruler la main " est la modification d'un paramètre en particulier de " La main ". Quel est ce paramètre ?

La température

Quelle est la valeur de " La température " avant l'execution de " Bruler la main " ?

basse

Quelle est la valeur de " La température " apres l'execution de " Bruler la main " ?

haute

Quelle est la cause de " La température " " haute " ?

cela est la CONSEQUENCE de la réalisation d'une fonction utile

Quelle est la fonction utile qui n'est pas assurée si " La température " n'est pas " haute " ?

Souder avec précision

Il faut donc résoudre la contradiction suivante: " La température " doit être " haute " afin d'assurer la réalisation de " Souder avec précision ", mais " La température " ne doit pas être " haute " afin d'éviter l'apparition de " Bruler la main ".

Nous allons analyser le système responsable de l'apparition de " Bruler la main ". Donnez un nom à ce système.

La pointe

Lors de la réalisation de " Bruler la main ", un élément de " La pointe " est en contact direct, agit directement sur " La main ". Quel est l'élément en question ?

Le manche

Quel paramètre de " Le manche " peut-on modifier afin d'empêcher l'apparition " Bruler la main " ?

Pouvoir isolant

Quelle valeur " Pouvoir isolant " de " Le manche " permet d'empêcher l'apparition de " Bruler la main " ?

Fort

Si " Pouvoir isolant " de " Le manche " est à " Fort ", le problème est-il résolu ?

Non

Quelle est la valeur actuelle pour " Pouvoir isolant " de " Le manche " ?

Faible

Si nous ne pouvons le changer, c'est que le fait d'avoir " Pouvoir isolant " de " Le manche " à " Faible " permet de réaliser une action utile. Quelle est cette fonction ?

Le coût du matériau

La contradiction suivante est alors à résoudre: " Pouvoir isolant " de " Le manche " doit être " Faible " pour assurer la réalisation de " Le coût du matériau ", mais " Pouvoir isolant " de " Le manche " doit être " Fort " pour " Bruler la main ".

Donnez un nom au projet.

Probleme du fer

Quel est l'objectif de l'etude ?

Amelioration d'un systeme existant.

Quel est le nom du systeme a ameliorer ?

Le fer

Quelle est la fonction principale réalisée par " Le fer " ?

Souder

De quelle maniere souhaitez-vous ameliorer le systeme " Le fer " ?

Suppression d'un effet nefaste.

Quel est l'effet genant, la fonction genante, a supprimer ? (formulation conseillée: un verbe + un COD)

Bruler l'utilisateur

La realisation d'une fonction s'effectue dans le but de modifier une caracteristique d'un objet. Quel est l'objet en question, pour la fonction " Bruler l'utilisateur " ?

La main de l'usager

" La main " est-il : (Substance, Systeme)

Substance

" Bruler la main " est la modification d'un paramètre en particulier de " La main de l'usager ". Quel est ce paramètre ?

Sa chaleur

Quelle est la valeur de " Sa chaleur " avant l'execution de " Bruler l'utilisateur " ?

faible

Quelle est la valeur de " Sa chaleur " apres l'execution de " Bruler l'utilisateur " ?

haute

Quelle est la cause de " Sa chaleur " " haute " ?

cela est la CONSEQUENCE de la realisation d'une fonction utile

Quelle est la fonction utile qui n'est pas assurée si " La température " n'est pas " haute " ?

La précision

Il faut donc résoudre la contradiction suivante: " Sa chaleur " doit être " haute " afin d'assurer la réalisation de " La précision ", mais " Sa chaleur " ne doit pas être " haute " afin d'éviter l'apparition de " Bruler l'utilisateur ".

Nous allons analyser le système responsable de l'apparition de " Bruler l'utilisateur ".
Donnez un nom à ce système.

Le système de chauffage du fer

Lors de la réalisation de " Bruler l'utilisateur ", un élément de " Le système de chauffage du fer " est en contact direct, agit directement sur " La main ". Quel est l'élément en question ?

Le manche

Quel paramètre de " Le manche " peut-on modifier afin d'empêcher l'apparition " Bruler l'utilisateur " ?

La longueur

Quelle valeur " La longueur " de " Le manche " permet d'empêcher l'apparition de " Bruler l'utilisateur " ?

Long

Si " La longueur " de " Le manche " est à " Long ", le problème est-il résolu ?

Non

Quelle est la valeur actuelle pour " La longueur " de " Le manche " ?

Normale

Si nous ne pouvons le changer, c'est que le fait d'avoir " La longueur " de " Le manche " à " Normale " permet de réaliser une action utile. Quelle est cette fonction ?

La précision

La contradiction suivante est alors à résoudre: " La longueur " de " Le manche " doit être " Normale " pour assurer la réalisation de " La précision ", mais " La longueur " de " Le manche " doit être " Long " pour " Bruler l'utilisateur ".

Donnez un nom au projet.

Rabah

Quel est l'objectif de l'étude ?

Amélioration d'un système existant.

Quel est le nom du système à améliorer ?

soudeur

Quelle est la fonction principale réalisée par " Le fer " ?

Souder

De quelle manière souhaitez-vous améliorer le système " Le fer " ?

Suppression d'un effet néfaste.

Quel est l'effet gênant, la fonction gênante, à supprimer ? (formulation conseillée: un verbe + un COD)

Bruler la main

La réalisation d'une fonction s'effectue dans le but de modifier une caractéristique d'un objet. Quel est l'objet en question, pour la fonction " Bruler l'utilisateur " ?

La main

" La main " est-il : (Substance, Système)

Substance

" Bruler la main " est la modification d'un paramètre en particulier de " La main de l'utilisateur ". Quel est ce paramètre ?

l'état

Quelle est la valeur de " l'état " avant l'exécution de " Bruler la main " ?

normal

Quelle est la valeur de " l'état " après l'exécution de " Bruler la main " ?

brule

Quelle est la cause de " la main " " brule " ?

cela est la CONSEQUENCE de la réalisation d'une fonction utile

Quelle est la fonction utile qui n'est pas assurée si " la main " n'est pas " brule " ?

Souder

Il faut donc résoudre la contradiction suivante: " l'état " doit être " brulé " afin d'assurer la réalisation de " Souder ", mais " l'état " ne doit pas être " brulé " afin d'éviter l'apparition de " Bruler la main ".

Donnez un nom au projet.

Fer a souder

Quel est l'objectif de l'étude ?

Amélioration d'un système existant.

Quel est le nom du système à améliorer ?

Fer a souder

Quelle est la fonction principale réalisée par " Fer a souder " ?

Souder des pièces ensemble par fusion

De quelle manière souhaitez-vous améliorer le système " Fer a souder " ?

Amélioration d'une fonction utile.

Quelle est la fonction dont vous souhaitez améliorer la réalisation ?

Souder avec précision

Cette fonction est-elle décomposable ?

Non

Parmi les fonctions suivantes, quelle est celle dont la réalisation pose le plus de problèmes ?

Souder avec précision

La réalisation d'une fonction s'effectue dans le but de modifier une caractéristique d'un objet. Quel est l'objet en question, pour la fonction " Souder avec précision " ?

La soudure

"La soudure " est-il : (Substance, Système)

Substance

"Souder avec précision " est la modification d'un paramètre en particulier de " la soudure ".
Quel est ce paramètre ?

La précision

Quelle est la valeur de " La précision " avant l'exécution de " Souder avec précision " ?

Nulle

Quelle est la valeur de " La précision " après l'exécution de " Souder avec précision " ?

Bonne

Il faut améliorer la réalisation de " Souder avec précision ". Quelle serait la valeur idéale du paramètre " La précision " de " La soudure " après la réalisation de " Souder avec précision " ?

##Retour en arrière, la fonction à améliorer n'est pas Souder avec précision...##

Quelle est la fonction dont vous souhaitez améliorer la réalisation ?

Tenir le fer pres de la pointe

Cette fonction est-elle décomposable ?

Non

Parmi les fonctions suivantes, quelle est celle dont la réalisation pose le plus de problèmes ?

Tenir le fer pres de la pointe

La réalisation d'une fonction s'effectue dans le but de modifier une caractéristique d'un objet. Quel est l'objet en question, pour la fonction " Tenir le fer pres de la pointe " ?

Fer a souder

" Tenir le fer pres de la pointe " est la modification d'un paramètre en particulier de " Fer a souder ". Quel est ce paramètre ?

La distance entre la main et la pointe

Quelle est la valeur de " La distance entre la main et la pointe " avant l'exécution de " Tenir le fer pres de la pointe " ?

Longue

Quelle est la valeur de " La distance entre la main et la pointe " après l'exécution de " Tenir le fer pres de la pointe " ?

Courte

Il faut améliorer la réalisation de " Tenir le fer pres de la pointe ". Quelle serait la valeur idéale du paramètre " La distance entre la main et la pointe " de " Fer a souder " après la réalisation de " Tenir le fer pres de la pointe " ?

Nulle

Quel paramètre de " Fer à souder " faut-il modifier afin d'avoir " La distance entre la main et la pointe " à la valeur " Nulle " ?

La longueur du manche

Quelle est la valeur de " La longueur du manche " de " Fer à souder " qui permet de mettre " La distance entre la main et la pointe " à " Nulle " ?

Petite

Si " La longueur du manche " de " Fer à souder " vaut " Petite ". Cela crée-t-il un effet gênant ?

Oui

Si on considère que l'effet gênant est la réalisation d'une fonction non souhaitée. Quelle est cette fonction ?

La main est brûlée

Il faut donc résoudre la contradiction suivante: " La longueur du manche " de " Fer à souder " doit être " Petite " afin d'avoir " La distance entre la main et la pointe " de " Fer à souder " à " Nulle ". Mais " La longueur du manche " de " Fer à souder " ne doit pas être " Petite " pour éviter l'apparition de " La main est brûlée ".

ANNEXE-XI. SECONDE SERIE D'EXPERIMENTATIONS

Etudiants dans leur 3^{ème} séance de 3 heures de projet TRIZ. Première consacrée à l'identification d'un projet sur lequel travailler, deuxième séance consacrée à la formulation du problème.

Edouard : piercing pour la langue

Le piercing est constitué d'une tige filetée auxquelles sont vissées des boules, qui se dévissent et sont avalées.

La formulation atteinte est : une CT entre la fixation et l'usinage de la tige filetée et une CP sur la démontabilité de la fixation.

L'utilisation du logiciel débouche sur la formulation d'une CP plus spécifique, portant sur la « dévissabilité ».

Antony : potence d'horlogerie pour démontage de montres

L'utilisation permet d'identifier les connaissances manquantes vis-à-vis du sujet. En fait, l'étudiant n'a pas de problèmes, dans l'état actuel des connaissances. Il possède des solutions potentiellement applicables, mais ne sait pas si elles le sont réellement ou non. Il doit en déterminer la viabilité sur le terrain, et le cas échéant identifier les causes de non applicabilité.

Emmanuel : système de serrage de pièce en bois massif sur CN.

A une connaissance moindre du projet.

L'utilisation du soft permet la mise en évidence des connaissances manquantes et à collecter. Dans un premier temps, le problème à été formulé sur une fonction correctement satisfaite, l'étudiant s'en est rendu compte et a du modifier le positionnement.

Le résultat obtenu est une formulation de contradiction plus générique que pour Yvon, mais du au manque de connaissances spécifiques.

Yvon : système de serrage de pièce en bois massif sur CN.

Positionnement par ventouses, trop grosses pour les pièces en bois de petites dimensions.

Il existe différentes valeurs de largeur et d'épaisseur de pièces, le démontage et le réglage sont manuels.

Jusqu'à présent ont été identifiés les problèmes rencontrés et les contraintes, six contradictions techniques ont été formulées.

L'utilisation du soft a permis d'identifier une CP propre et soluble, dans une voie qui n'avait pas été identifiée préalablement.

EDOUARD

Donnez un nom au projet.

piercing pour la langue

Quel est l'objectif de l'étude ?

Amélioration d'un système existant.

Quel est le nom du système à améliorer ?

système de fixation du piercing

Quelle est la fonction principale réalisée par " système de maintien des pièces " ?

tenir le piercing

De quelle manière souhaitez-vous améliorer le système " système de maintien en position de la montre " ?

Suppression d'un effet néfaste.

Quel est l'effet gênant, la fonction gênante, à supprimer ? (formulation conseillée: un verbe + un COD)

devissement impromptu

La réalisation d'une fonction s'effectue dans le but de modifier une caractéristique d'un objet. Quel est l'objet en question, pour la fonction " devissement impromptu " ?

Le bijou

" Le bijou " est-il : (Substance, Système)

Substance

" devissement impromptu " est la modification d'un paramètre en particulier de " Le bijou ". Quel est ce paramètre ?

La fixation

Quelle est la valeur de " La fixation " avant l'exécution de " réglage des verins " ?

immobile

Quelle est la valeur de " La fixation " après l'exécution de " réglage des verins " ?

absente

Quelle est la cause de " La fixation " " absente " ?

cela est la CONSEQUENCE de la réalisation d'une fonction utile

Quelle est la fonction utile qui n'est pas assurée si " la localisation " n'est pas " absente " ?

interchangeabilité du bijou

Il faut donc résoudre la contradiction suivante: " La fixation " doit être " absente " afin d'assurer la réalisation de " interchangeabilité du bijou ", mais " La fixation " ne doit pas être " absente " afin d'éviter l'apparition de " dévissage impromptu ".

Nous allons analyser le système responsable de l'apparition de " dévissage impromptu ".
Donnez un nom à ce système.

fixation

Lors de la réalisation de " dévissage impromptu ", un élément de " fixation " est en contact direct, agit directement sur " Le bijou ". Quel est l'élément en question ?

Le filetage

Quel paramètre de " Le filetage " peut-on modifier afin d'empêcher l'apparition " dévissage impromptu " ?

devissabilité

Quelle valeur " devissabilité " de " Le filetage " permet d'empêcher l'apparition de " dévissage impromptu " ?

nulle

Si " devissabilité " de " Le filetage " est à " nulle ", le problème est-il résolu ?

Non

Quelle est la valeur actuelle pour " devissabilité " de " Le filetage " ?

existante

Si nous ne pouvons le changer, c'est que le fait d'avoir " devissabilité " de " Le filetage " à " existante " permet de réaliser une action utile. Quelle est cette fonction ?

interchangeabilité du bijou

La contradiction suivante est alors à résoudre: " devissabilité " de " Le filetage " doit être " existante " pour assurer la réalisation de " interchangeabilité du bijou ", mais " devissabilité " de " Le filetage " doit être " nulle " pour " dévissage impromptu ".

ANTONY

Donnez un nom au projet.

potence horlogère

Quel est l'objectif de l'étude ?

Amélioration d'un système existant.

Quel est le nom du système à améliorer ?

système de maintien en position de la montre

Quelle est la fonction principale réalisée par " système de maintien en position de la montre " ?

maintien en position du boîtier de montre

De quelle manière souhaitez-vous améliorer le système " système de maintien en position de la montre " ?

Amélioration d'une fonction utile.

Quelle est la fonction dont vous souhaitez améliorer la réalisation ?

maintien en position du boîtier de montre

Cette fonction est-elle décomposable ?

Oui

Définissez une sous-fonction de "maintien en position du boîtier de montre".

mise en position du boîtier

Y'a-t-il d'autres sous-fonctions de "maintien en position du boîtier de montre"?

Oui

Définissez une sous-fonction de "maintien en position du boîtier de montre".

maintien du boîtier

Y'a-t-il d'autres sous-fonctions de "maintien en position du boîtier de montre"?

Non

Parmi les fonctions suivantes, quelle est celle dont la réalisation pose le plus de problèmes ?

mise en position du boîtier

La réalisation d'une fonction s'effectue dans le but de modifier une caractéristique d'un objet. Quel est l'objet en question, pour la fonction " mise en position du boîtier " ?

le boîtier

"le boîtier " est-il : (Substance, Systeme)

Substance

" mise en position du boîtier " est la modification d'un paramètre en particulier de " le boîtier ". Quel est ce paramètre ?

la position

Quelle est la valeur de " la position " avant l'exécution de " mise en position du boîtier " ?

aléatoire

Quelle est la valeur de " la position " après l'exécution de " mise en position du boîtier " ?

centrée par rapport à l'axe de la potence

Il faut améliorer la réalisation de " mise en position du boîtier ". Quelle serait la valeur idéale du paramètre " la position " de " le boîtier " après la réalisation de " mise en position du boîtier " ?

centrée par rapport à l'axe de la potence

Quel paramètre de " le boîtier " faut-il modifier afin d'avoir " la position " à la valeur " centrée par rapport à l'axe de la potence " ?

Le boîtier n'est pas modifiable, la fonction à améliorer ne porte pas sur le boîtier, mais sur l'outil qui réalise la mise en position du boîtier. C'est à dire les ergots.

« réglage des ergots de maintien »

« les ergots », « la position », de « centré et éloigné de l'axe » à « centré et au contact du boîtier », valeur_objectif = « centré et au contact du boîtier automatiquement ».

« automatiser le déplacement des ergots »

« système de réglage », « automatisation », de « non existant » à « non existant »,
valeur_objectif = « existant ».

Quel paramètre de " système de réglage " faut-il modifier afin d'avoir " automatisation " à
la valeur " existant " ?

le dynamisme

Quelle est la valeur "le dynamisme" de " système de réglage " qui permet de mettre "
automatisation " a " existant " ?

grand

Si "le dynamisme" de " système de réglage " vaut "grand". Cela crée-t-il un effet gênant ?

Non

Le problème est le suivant: il faut améliorer la réalisation de "automatiser le déplacement
des ergots", mais vous ne savez pas comment faire. Utilisez les lois d'évolution des systèmes
techniques avec le système identifié.

EMMANUEL

Donnez un nom au projet.

serrage des elements en bois pour CN

Quel est l'objectif de l'etude ?

Amelioration d'un systeme existant.

Quel est le nom du systeme a ameliorer ?

système de maintien des pieces en bois

Quelle est la fonction principale réalisée par " système de maintien en position de la montre " ?

maintien des pieces en bois

De quelle maniere souhaitez-vous ameliorer le systeme " système de maintien en position de la montre " ?

Amelioration d'une fonction utile.

Quelle est la fonction dont vous souhaitez améliorer la réalisation ?

mise en position des pieces

Cette fonction est-elle décomposable ?

Non

Parmi les fonctions suivantes, quelle est celle dont la réalisation pose le plus de problèmes ?

mise en position des pieces

La realisation d'une fonction s'effectue dans le but de modifier une caracteristique d'un objet. Quel est l'objet en question, pour la fonction " mise en position des pieces " ?

Les 5 pieces a usiner

" Les 5 pieces a usiner " est-il : (Substance, Systeme)

Substance

" mise en position des pieces " est la modification d'un paramètre en particulier de " Les 5 pieces a usiner ". Quel est ce paramètre ?

positionnement dans l'espace

Quelle est la valeur de " positionnement dans l'espace " avant l'exécution de " mise en position des pièces " ?

aléatoire

Quelle est la valeur de " positionnement dans l'espace " après l'exécution de " mise en position des pièces " ?

fixe, a precision donnee

Il faut améliorer la réalisation de " mise en position des pièces ". Quelle serait la valeur idéale du paramètre " positionnement dans l'espace " de " Les 5 pièces à usiner " après la réalisation de " mise en position des pièces " ?

precis par rapport aux tetes d'usinage

Quel paramètre de " Les 5 pièces à usiner " faut-il modifier afin d'avoir " positionnement dans l'espace " à la valeur " précis par rapport aux têtes d'usinage " ?

Les 5 pièces à usiner ne sont pas modifiable, la fonction à améliorer ne porte pas sur les pièces, mais sur l'outil qui réalise la mise en position des pièces. C'est à dire les butées.

« réglage des butées »

« les butées », « positionnement dans l'espace », de « approximativement connue » à « position connue », valeur_objectif = « égales à celles de la partie commande ».

Quel paramètre de " les butées " faut-il modifier afin d'avoir " positionnement dans l'espace " à la valeur " égales à celles de la partie commande " ?

le déplacement en Y

Quelle est la valeur "le dynamisme" de " système de réglage " qui permet de mettre " automatisation " à " existant " ?

libre et connu

Si " le déplacement en Y " de " les butées " vaut " libre et connu". Cela crée-t-il un effet gênant ?

Oui

Si on considère que l'effet gênant est la réalisation d'une fonction non souhaitée. Quelle est cette fonction ?

définir l'origine de la pièce

Il faut donc résoudre la contradiction suivante: " le déplacement en Y " de " les butées " doit être " libre et connu" afin d'avoir " positionnement dans l'espace " de " les butées " a " égales à celles de la partie commande ". Mais " le déplacement en Y " de " les butées " ne doit pas être " libre et connu" pour éviter l'apparition de " définir l'origine de la pièce ".

YVON

Donnez un nom au projet.

Systeme de serrage des elements en bois massif pour commande numerique

Quel est l'objectif de l'etude ?

Amelioration d'un systeme existant.

Quel est le nom du systeme a ameliorer ?

systeme de maintien des pieces

Quelle est la fonction principale réalisée par " système de maintien des pieces " ?

maintenir les pieces

De quelle maniere souhaitez-vous ameliorer le systeme " système de maintien en position de la montre " ?

Amelioration d'une fonction utile.

Quelle est la fonction dont vous souhaitez améliorer la réalisation ?

reglage des verins

Cette fonction est-elle décomposable ?

Non

Parmi les fonctions suivantes, quelle est celle dont la réalisation pose le plus de problèmes ?

reglage des verins

La realisation d'une fonction s'effectue dans le but de modifier une caracteristique d'un objet. Quel est l'objet en question, pour la fonction " reglage des verins " ?

les verins

"les verins " est-il : (Substance, Systeme)

Substance

"reglage des verins " est la modification d'un paramètre en particulier de " les verins ".
Quel est ce paramètre ?

la localisation

Quelle est la valeur de " la localisation " avant l'exécution de " réglage des verins " ?

present

Quelle est la valeur de " la localisation " après l'exécution de " réglage des verins " ?

non present

Il faut améliorer la réalisation de " réglage des verins ". Quelle serait la valeur idéale du paramètre " la localisation " de " les verins " après la réalisation de " réglage des verins " ?

au contact de la piece

Quel paramètre de " les verins " faut-il modifier afin d'avoir " la localisation " à la valeur " au contact de la piece " ?

la course des verins

Quelle est la valeur " la course des verins " de " les verins " qui permet de mettre " la localisation " à " au contact de la piece " ?

grande

Si " la course des verins " de " les verins " vaut "grande". Cela crée-t-il un effet gênant ?

Oui

Si on considère que l'effet gênant est la réalisation d'une fonction non souhaitée. Quelle est cette fonction ?

positionner le verin

Il faut donc résoudre la contradiction suivante: " la course des verins " de " les verins " doit être "grande" afin d'avoir " la localisation " de " les verins " à "grande". Mais " la course des verins " de " les verins " ne doit pas être "grande" pour éviter l'apparition de " positionner le verin ".