

INTRODUCTION DE TECHNIQUES D'INTELLIGENCE ARTIFICIELLE DANS LES EQUIPES RECHERCHE DE LA CHARGE

I - CONSIDERATIONS GENERALES

a) avantages :

L'emploi de techniques d'IA (Intelligence Artificielle) repose sur quatre différences essentielles avec l'informatique classique telle qu'elle est pratiquée dans la charge (Fortran + Générateur d'écran sur micro MS/DOS et Fortran sur DEC VMS)

* utilisation de machines plus puissantes et comportant une interface homme/machine plus évoluée sous des systèmes d'exploitation différents (UNIX ou LISP) ces stations de travail permettent de s'affranchir des limites traditionnelles de MS DOS et de fournir un environnement de développement de logiciel plus productif car en même temps plus efficace et plus agréable à utiliser.

* utilisation de langages de programmation plus puissants (C et LISP qui permettent une programmation plus aisée et plus maintenable pour des problèmes donc plus complexes (il y a autant de différence entre LISP et FORTRAN qu'entre FORTRAN et l'assembleur du point de vue de la facilité de programmation pour résoudre un problème complexe).

* utilisation de concepts logiciels avancés permettant la résolution des problèmes complexes grâce à des outils qui ont été mis au point et testés dans le cadre de ces problèmes complexes (ex : moteur d'inférence, base de connaissance, langage orienté objet...).

* approche des problèmes radicalement différente d'une approche classique : analyse fonctionnelle, programmation, déboggage, modification ; à cette approche succède une approche centrée sur un prototypage continu c'est-à-dire qu'au lieu d'étudier au préalable la forme définitive que doit prendre le produit et procéder à une analyse fonctionnelle hexaustive qui peut se révéler impossible à effectuer et qui donne une grande rigidité au projet, on procède à

l'implémentation de petites fonctionnalités qui grâce à un environnement adequat peuvent être testées immédiatement permettant un développement en temps réel basé sur un monitoring (feed back auto-correctif) de la part de l'expert ou du futur utilisateur. Cette démarche typique des systèmes experts permet de palier l'impossibilité d'analyse fonctionnelle préalable.

b) Contraintes

La mise en oeuvre de ces techniques n'est pas gratuite, elle implique quatre contraintes dont il faut être conscient :

* premièrement, il est nécessaire d'acheter des machines mono-utilisateur d'un coût unitaire élevé (entre 100 KF (Sun 50) et 600 KF (Sun 4 ou Symbolics)).

* deuxièmement, il faut savoir qu'on ne pourra utiliser les concepts de l'IA de manière efficace que si on dispose de l'environnement logiciel adequat. Celui-ci est constitué d'ateliers logiciels permettant la programmation directement dans des langages spécialisés de plus haut niveau que le LISP (langage orienté objet + mécanismes particuliers) et facilitant la boucle test-modification. Ces environnements coûtent entre 150 KF (Le LISP + MPLRO) et 350 KF (KEE ou Knowledge Craft).

* troisièmement, ces technologies ne peuvent être mises en oeuvre au niveau conception que par des informaticiens de haut niveau ayant une formation en intelligence artificielle (niveau DEA ou grande école scientifique). Néanmoins, l'utilisation des produits ainsi conçus correctement interfacés (c'est-à-dire avec des menus et des "help" en lignes) ne requérera pas de compétence particulière si ce n'est une initiation pratique au lancement des programmes de la machine considérée (une 1/2 heure de formation).

* quatrièmement, la démarche particulière de conception requière une participation constante tout au long du projet de spécialistes de la discipline qui doivent tester les prototypes intermédiaires et suggérer des modifications, le responsable du projet essayant alors de formaliser la remarque du spécialiste en discutant avec lui jusqu'à complète compréhension du problème. Cette démarche connue en informatique classique durant l'analyse fonctionnelle est mise en

oeuvre lors de la conception d'un système expert avec une toute autre intensité (l'implication d'experts peut atteindre 50 % de leur temps durant 1 an).

c) Nécessités

L'évolution des marchés tend à montrer que les opérateurs se munissent d'outils de plus en plus évolués qui permettent des décisions de plus en plus rapides. Par contre-coup, les marchés viennent de plus en plus volatils renforçant ainsi l'importance du timing dans le trading et obligeant à une gestion du risque de plus en plus rigoureuse et complexe.

D'autre part, la libéralisation de 92 nous amène à nous préparer à des niveaux de compétition jamais atteint par le passé. L'observation des méthodes des futurs concurrents nous amène à nous interroger sur l'opportunité de ces techniques. De nombreuses expériences de par le monde ont montré l'applicabilité de ces techniques au milieu industriel (nombreux systèmes experts réalisés dans des domaines variés). Certaines expériences menées par des pionniers dans le milieu financier (trader's assistant...) nous montrent que ces techniques vont bouleverser les habitudes de fonctionnement des opérateurs qui vont être capable de réagir rapidement et de manière appropriée aux évènements mais aussi d'effectuer des analyses complexes de l'ensemble du marché (scanning intelligent) très rapidement (quelques minutes à quelques heures), le comportement des marchés va s'en trouver modifié, tout opérateur ne maîtrisant pas ces techniques se trouvera en position d'infériorité dans le trading.

II - LES PROJETS POSSIBLES DANS LE DEPARTEMENT OPTIONS

Jusqu'à présent, trois projets ont été formulés. Il est certain que dans le futur, le développement d'un projet avançant d'autres idées naîtront qui amèneront soit des redélimitations des projets existant soit des projets complémentaires.

1) Premier projet : un atelier de chartisme

a) le but de cet atelier serait de pouvoir mettre au point et tester des nouveaux indices et de nouvelles règles de trading relatif, à ces indices. Ces règles

de trading pourront incorporer des considérations chartistes de formes caractéristiques comme le "tête-épaule" ce qui impliquera des capacités de reconnaissance des formes et une quantification de nature probabiliste des objectifs de prix. Ce système pourra donc aussi être utilisé pour faire du scanning des marchés et servir de base à un futur système expert de synthèse automatique de l'état des marchés.

Le projet est important mais un premier prototype opérationnel doit être terminé en 6 - 9 mois.

b) l'avantage de ce système sera la mise au point d'indices opérationnels (optimisation statistique des paramètres et règles de trading associées) susceptibles d'être inclus dans le produit chartiste sur micro. De même le scanning intelligent qu'il permettra aidera considérablement à la prise de position initiale des traders en début de journée. Ces indices pourraient aussi être exploités commercialement pour leur intérêt propre.

c) les contraintes de ce système sont :

- * 6 - 9 mois de travail pour une équipe de 2 personnes à 80 % de charge ;
- * un environnement logiciel et matériel (SUN + LISP + AIDA) déjà en place à la charge + la fourniture de cours de valeurs sur 3 ans avec la meilleure précision possible (plusieurs cours par jour + volumes) sur SUN par un moyen à étudier.

2) Aide au trading

a) Ce projet s'inspire de celui qui a été réalisé par Arthur D. Little (Cabinet Américain de consultants) pour la Security Pacific Bank (CA). Il s'agit d'un système Expert emmagasinant la connaissance des marchés et des opérateurs susceptibles d'apprécier les nouvelles données en temps réel par Reuter ou Antiope de même que les rumeurs d'OPA ou de réunions des ministres des finances des pays occidentaux, ce système donne des conseils d'achat et de vente ou de spreads pour le market making. Il s'agit d'un grand projet pouvant s'étaler sur deux ans pour la réalisation d'un système opérationnel. Il devra être spécialisé sur un marché particulier, plus la spécialisation sera poussée, plus les chances de réussite du projet seront grandes.

b) L'avantage de ce système est bien sûr une plus grande rapidité de réaction des opérateurs et une meilleure adéquation de leur réaction à la situation particulière permettant de réagir à chaud avec un comportement que l'on aurait eu à froid si on avait eu le temps de la réflexion.

c) contraintes : il s'agit typiquement d'un grand système expert, les contraintes sont :

* d'une part logicielle et matérielle :

- une SUN plus puissante (3/60 ou 3/260 ou 4 ou une SYMBOLICS (400 KF)
- un logiciel générateur de système expert à la mesure du problème (350 KF pour KEE, Knowledge Craft ou SMECI) ;

* d'autre part organisationnelle :

- 2 années hommes d'ingénieur de la connaissance (spécialiste de la réalisation du système expert) ;
- 2 années hommes de traders expérimentés des marchés en question.

La charge très lourde en personnel très qualifié est le trait caractéristique de ce genre de système expert orienté vers une application très pragmatique comportant un grand nombre de règles d'expertise.

3) Aide à l'Ingénierie financière

a) il s'agit d'un projet de simulation du marché financier avec toute la puissance des nouveaux concepts de simulation par système . L'idée est de formaliser un générateur de scénario qui permettent de tester des constructions d'ingénierie financière dans un grand nombre de cas possibles, tels qu'ils peuvent être envisagés dans les circonstances de l'étude. La différence avec le premier projet dans lequel des règles de trading peuvent être testées sur l'avenir est que dans le cas du premier projet le module de simulation de scénario sera rajouté ensuite. Le troisième projet servira à tester des constructions financières plus complexes et sur des scénarios plus élaborés que dans le cas du module faisant partie du premier projet.

b) l'avantage de ce système est de permettre de se faire une idée précise sur les pertes et profits des différents services que l'on peut offrir à la clientèle et par suite tarifer ces services de manière plus rigoureuse (spread, courtage,...). Par exemple, ce système nous permettra de simuler des politiques de couverture différentes de celle du Δ neutre.

c) contraintes : la réflexion portera essentiellement sur la modélisation du marché. Les instruments nécessaires sont donc un outil de modélisation conceptuelle (paradoxalement il s'agit des mêmes machines et mêmes logiciels que pour le projet 2) soit une machine 16 mega octet de mémoire centrale (300 KF) et le logiciel (KEE ou Knowledge Kraft) (350 KF).

La contrainte humaine est nettement moins lourde que dans le projet 2), car ce projet ne nécessite pas forcément une accumulation importante de connaissances particulières mais plutôt une réflexion globale.

4) Recommandation de sélection

Au vu des considérations précédentes, il apparaît que le projet 2), sauf demande particulière est un projet gourmant en temps de traders s'il peut s'avérer nécessaire dans l'avenir il n'est pas dans l'ordre des priorités vu la faiblesse en personnel des équipes de trading option. Le projet 3) suppose une activité importante d'ingénierie financière ; là encore il est possible que cette activité se développe dans l'avenir sur le schéma de l'étude Aigle-Azur ou de l'option sur indice mais ce n'est pas non plus une priorité du département. Le seul projet qui ait un intérêt immédiat est celui qui permettrait d'avoir de meilleures règles de trading et de disposer de meilleures indices afin d'améliorer la rentabilité de nos interventions sur le marché...

III - DESCRIPTION TECHNIQUE DU PROJET N° 1 ET PLANNING PROPOSE

1) Idées générales relatives à la réalisation du projet

Ce projet n'est pas classique au sens où s'il emploie des techniques d'IA connues et maîtrisées, il les emploiera dans une application peu explorée à ce jour en Europe. Il ne peut s'agir d'un projet de R et D. Nous y incorporons une

obligation de résultat dans la mesure où il s'agit d'un système qui démontrera la technologie. On se donne donc un délai de 9 mois pour parvenir à un système démontrable soit 4 mois d'implémentation, 4 mois de tests et un mois préalable qui n'est pas opérationnel mais plutôt une phase de transition et de formation.

2) Les différentes fonctionnalités du système

a) langage de spécification d'index (momentum, RSI,...) et d'outils de chartisme (tête-épaules, points et figures,...) cette fonctionnalité délimite la richesse des instruments chartistes qui seront examinés dans le système, la notion de langage implique une grande richesse dans les possibilités de formulation :

Name : indice X

$$Sortie = \frac{1}{1 + Volume quotidien} \times \frac{Plus haut du jour}{Plus bas du jour}$$

Y =

Name : tête-épaules
 X1 = canal haussier de largeur (I1)
 X2 = canal baissier de largeur (I2)
 X3 = canal haussier de largeur (I3)
 X4 = canal baissier de largeur (I4)
 X5 = canal haussier de largeur (I5)
 X6 = canal baissier de largeur (I6)
 connecté (X1, X2), connecté (X2, X3), connecté (X3, X4),
 connecté (X4, X5), connecté (X5, X6)
 sortie = sortie (X6)
 neckline = neckline (X3, X4)
 return max (I1, I2, I3, I4, I5, I6)

b) langage de spécification de règles de trading si :

Y1 = tête-épaule et S = Sortie - neckline
 et S/Y > 20 %

alors initier position longue = 1/10 disponibilité

c) le système possédant les points 1 et 2 nous permet alors de tester les règles de trading sur l'historique des cours et de voir si les règles de trading génèrent un bénéfice ou non.

d) langage de spécification de scénario d'évolution de cours sur la base d'une série de paramètres pouvant caractériser l'évolution des marchés dans les semaines ou les mois à venir, le système conçoit un grand nombre de scénarios différents d'évolution des marchés qui respectent la valeur de ces paramètres.

Ex :

CAC régulièrement baissier sur les six mois à venir de pente annuelle moyenne = 5 %

Probabilité d'accident haussier ou baissier
 $p = 0.01$ niveau = 10 %

Volatilité moyenne $\sigma = 20\% / \text{an}$
variation de la volatilité : hausse régulière 10 % /an

scénarios générés respectant ces hypothèses : 250

e) supposant donc les points 1, 2 et 4, le système serait capable de simuler le trading sur un grand nombre d'hypothèses de marchés possédant toutes les mêmes caractéristiques de tendance, de volatilité, etc... et donner une distribution statistique du profit réalisé à l'échéance considérée, ainsi qu'un certain nombre de paramètres nous intéressant (couverture maximum exigée, position débitrice maximale...)

f) les règles de trading peuvent comporter un certain nombre de paramètres :

Ex : p : pourcentage de la largeur du canal dont on sort, que l'on vient de franchir et à partir duquel une prise de position est effectuée.

Ce module nous permet d'enchaîner des simulations où l'on fait varier " p " et nous permet d'optimiser la valeur de " p " et de définir la robustesse des règles de trading à des variations de caractéristiques du marché.

Le projet est divisé grossièrement en **4 phases inégales** :

* La première phase ou **phase objet** vise à construire les objets et l'architecture de base du système : c'est une phase assez longue à cause de tous les problèmes liés au démarrage de l'activité d'IA.

* La deuxième phase est la **phase simulation** où sont implémentés les deux types de simulation à savoir la **simulation historique** et la **simulation sur scénario**.

* La troisième phase est dite **phase statistique** où les fonctionnalités d'optimisation statistiques des paramètres de trading sont implémentées. Au cours de cette phase, les fonctionnalités de scanning sont aussi implémentées.

* La dernière phase ou **phase finale** verra l'ajout d'aide en ligne, le renforcement de la robustesse du système, la rédaction d'une documentation d'utilisation et celle d'une note de synthèse opérationnelle précisant les résultats des différents tests qui ont été fait avec le système.

- Tâche 1** Implémentation du noyau de langage objet utilisé dans toute la suite du projet. Ce noyau supportera la métaphore des slots et des facettes de l'héritage de structure et de valeur (lien super) de la particularisation à partir du prototype (métaclasses : lien méta), des démons sur l'accès aux slots et sur l'instanciation, des méthodes de détermination attachées aux slots, de la gestion des slots relationnels doublement chaînés. Cette tâche verra l'écriture des primitives Lisp permettant la manipulation de ces objets.
- Tâche 2** Implémentation d'une architecture supportant les différents modules. Cette architecture s'inspirera de celle connue sous le nom de "Blackboard", le Blackboard sera représenté ici principalement par le marché et un portefeuille de positions sur lequel des règles de simulation viendront modifier ce dernier (règle d'exploration historique, règle de simulation de marché futur, règle de trading, règle statistiques d'optimisation). Un mécanisme d'agenda fournira toutes les fonctionnalités de gestion des différentes sources de connaissance et des fonctionnalités . Cette tâche verra l'écriture d'un chaîneur avant muni d' un pattern-matching avec variables ainsi que les primitives Lisp de gestion de l'agenda.
- Tâche 3** Implémentation du premier parser (programme qui transforme une description externe. Ex : texte fourni par l'utilisateur ou sélection à base de menus en une description interne en termes d'objets opérationnels du point de vue de l'interaction avec le Blackboard) il s'agit du parser chartiste. L'utilisateur rentre et nomme une ou plusieurs descriptions d'index ou de figures chartiste. Des prédictats de reconnaissance des formes sont utilisés permettant l'implémentation du RSI, des momentums et des moyennes, et canaux, triangles, tête-épaules sur les cours, momentums des points et figures. Cette tâche aboutira à la construction d'un certain nombre d'objets ordinaires et d'un éditeur d'objets spécialisés au travers duquel se manifestera le parser.
- Tâche 4** Cette tâche verra l'implémentation des prédictats de reconnaissance des formes utilisées dans la tâche précédente avec plusieurs méthodes différentes utilisables pour cette fonction.

- Tâche 5** Au cours de cette tâche le parser de règles de trading est construit. Ces règles de trading toutes synchrones décrivent le comportement d'un trader en face de certaines conditions d'évolution de marché et de conditions d'évolution de ses positions. Ce parser enregistre une série d'objets du type règle avec un prémissé et une conclusion mais aussi d'autres objets avec des méthodes spécifiques qui représentent les autres aspects du comportement : positions du trader, couvertures, profits, ... le résultat de cette tâche est un ensemble d'objets et un éditeur d'objets spécialisés.
- Tâche 6a** Le but du parser de simulation est la donnée au système de simulation des indications sur la manière d'effectuer sa tâche, à savoir quelle donnée enregistrer tout au long de la simulation, quelle donnée visualiser tout au long de la simulation, sur quelle donnée historique effectuer cette simulation, quand s'arrêter. Le résultat de cette tâche est ici aussi un ensemble d'objets nouveaux et un autre éditeur d'objets spécialisés.
- Tâche 6b** Le but de cette tâche est de s'assurer que le système aura à sa disposition, sous forme d'objets, les cours et volumes nécessaires. La résolution de tous les petits problèmes techniques sous-jacents et la réalisation d'un manager spécialisé clôturera cette tâche.
- Tâche 7** Le but de cette tâche est l'assemblage des différents éléments fabriqués dans les tâches précédentes pour générer une première démonstration des fonctionnalités de simulation historique, un contrôleur général AIDA sera écrit et une première version packagée.
- Tâche 8** Le but de cette tâche est de tester le système de simulation historique sur un certain nombre de cas et de recenser les manques d'expertise et définir ainsi les ajouts nécessaires pour réaliser un système utilisable.
- Tâche 9** Implantation du parser de scénario d'évolution. Ce parser comportera un générateur de chemin aléatoire respectant certaines contraintes globales comme la volatilité ou des contraintes sur cette volatilité (croissance de cette volatilité par exemple).

- Tâche 10** Implémentation de la fonctionnalité de simulation sur scénario. Cette fonctionnalité sera parfaitement intégrée dans l'outil permettant, par exemple, de générer par simulation historique des contraintes et générer, à partir de ces contraintes, des scénarios qui pourront tester la robustesse de ces contraintes.
- Tâche 11** Deuxième test opérationnel. On s'attachera, au cours de ces tests, à se faire une idée de la pertinence des scénarios générés, c'est-à-dire à savoir si la sélection des paramètres qui doivent représenter ces scénarios est correcte.
- Tâche 12a** Implémentation du parser statistique : ce parser permettra de tirer pleinement profit des simulations en spécifiant des méthodes d'optimisation statistiques de paramètres. Ce parser générera automatiquement les structures de contrôle qui permettront à l'optimiseur d'utiliser les simulateurs de la manière appropriée.
- Tâche 12b** Implémentation de la fonctionnalité d'optimisation paramétrique des règles de trading.
- Tâche 13** Troisième test opérationnel. Ces tests répondront à la question : les méthodes définies en 12a permettront-elles d'optimiser des règles de trading ? Si oui lesquelles ? (première approche grossière).
- Tâche 14** Implémentation de la fonctionnalité de génération de scanneur (petit parser intégré). Le système générera des exécutables Lisp capables de scanner le marché suivant certaines caractéristiques précisées lors de la génération. Les exécutables générés seront d'une très grande simplicité d'emploi.
- Tâche 15** Quatrième test opérationnel. Le but de ces tests est d'examiner l'utilité des scanners de marché dans le contexte du département voire de la charge et de répercer les points améliorables.
- Tâche 16a** Amélioration de la robustesse du système par l'intégration d'aide en ligne et la fermeture des accès système à l'utilisateur.

Tâche 16b Rédaction d'une documentation d'utilisation.

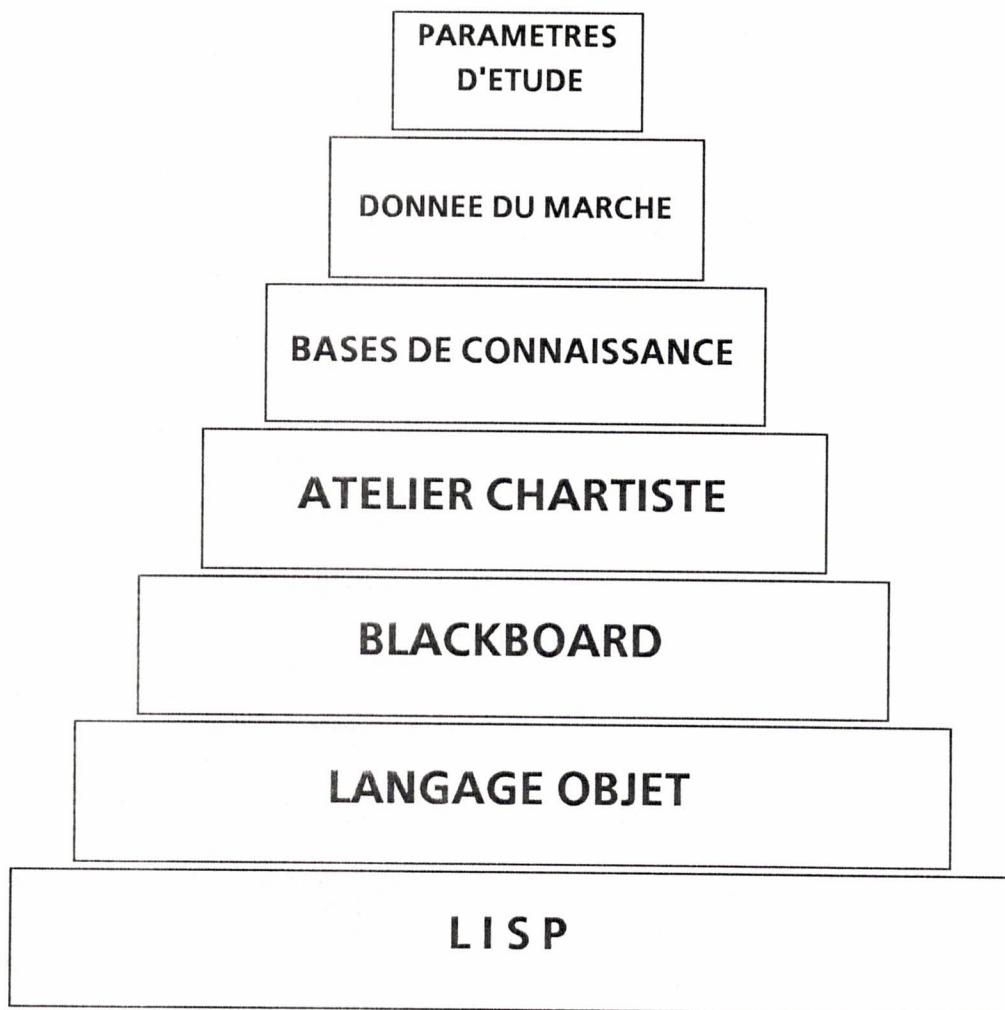
Tâche 16c Synthèse opérationnelle. Le but de cette tâche est d'effectuer une synthèse de tous les tests effectués lors des tests opérationnels, de noter les améliorations déjà apportées et les points encore améliorables, enfin de donner une conclusion sur le projet et sur la manière dont il a été mené.

DELAIS DE REALISATION

Les délais donnés ci-dessous sont des délais courts au sens où ils ne permettent pas la réalisation d'un système commercialisable et extensible à souhait. Ils sont longs au sens où ils occuperont une équipe de deux personnes et aboutiront à la formation d'une des personnes (les techniques employées diffèrent beaucoup des techniques informatiques classiques et leur apprentissage est plus long).

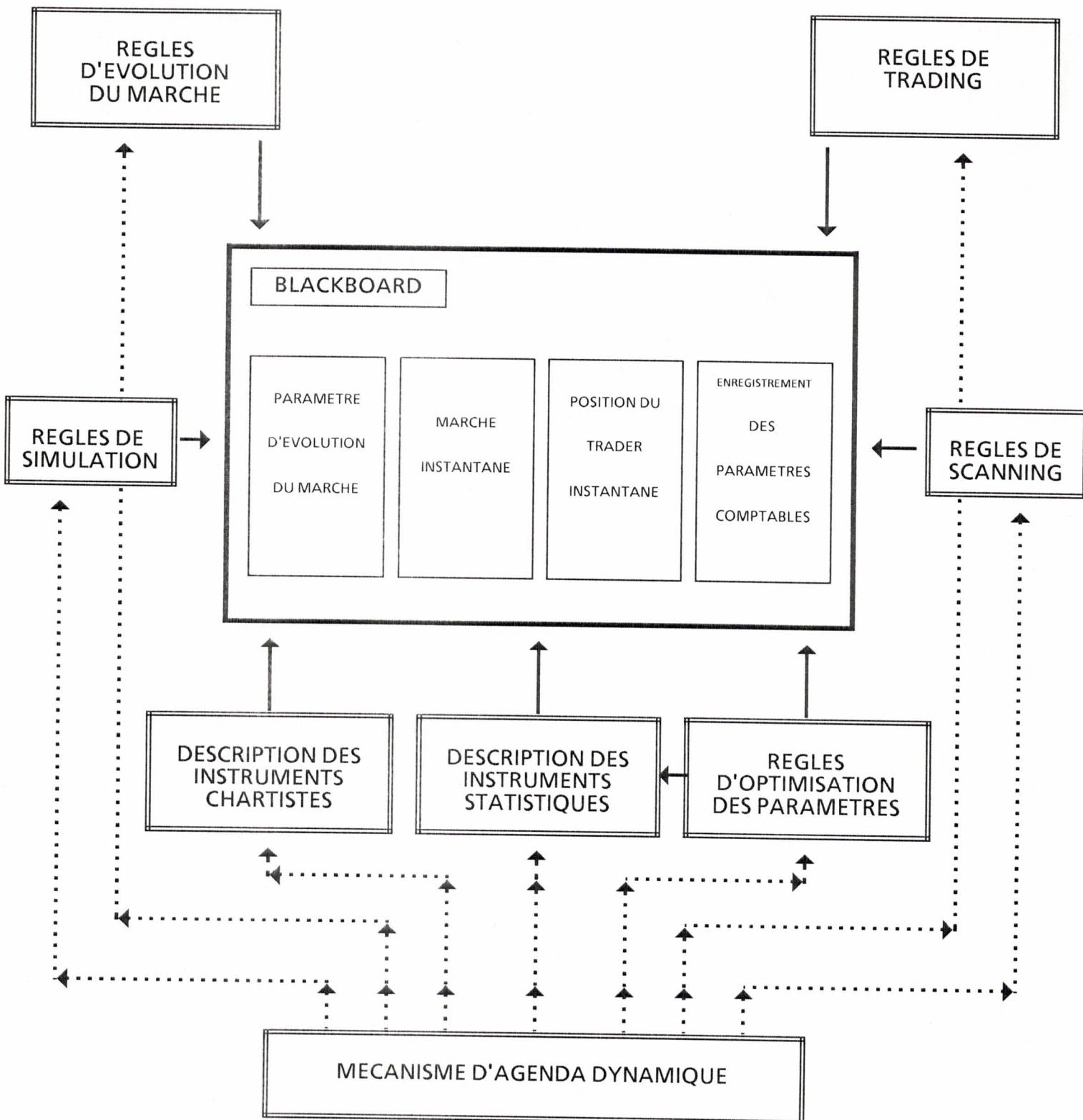
TACHE	DELAI (sem)	TACHE	DELAI (sem)
1	3	10	1
2	3	11	1
3	2	12a	2
4	2	12b	2
5	3	13	1
6a	1	14	2
6b	1	15	1
7	2	16a	1
8	1	16b	2
9	2	16c	1

ARCHITECTURE DU SYSTEME EN COUCHE



Note : Chaque couche supérieure utilise la couche inférieure pour son codage.

ARCHITECTURE FONCTIONNELLE



Note : le mécanisme d'agenda n'est pas un superviseur car il est "data-driven".

