

# VERS UNE INFRASTRUCTURE D'INTEGRATION POUR LE DÉVELOPPEMENT DE SYSTEMES D'AIDE A LA DECISION ENVIRONNEMENTALE

J. SERMENT, B. ESPINASSE et E. TRANVOUEZ

LSIS (UMR CNRS 6168)  
Université Paul Cézanne – Aix Marseille III  
Domaine Universitaire de Saint-Jérôme  
Avenue Escadrille Normandie-Niemen  
13397 Marseille Cedex 20 - France

{julien.serment, bernard.espinasse, erwan.tranvouez}@lsis.org

**RÉSUMÉ :** *La complexité des dynamiques environnementales nécessite le développement d'outils informatisés puissants d'aide à la décision ou Systèmes d'Aide à la Décision Environnementale (SADE). Cette recherche a pour objectif de faciliter le développement de tels systèmes. Une analyse comparative de plusieurs projets de SADE permet d'identifier les fonctionnalités majeures que peuvent offrir de tels systèmes pour leur(s) utilisateur(s) décideur(s). Après un inventaire détaillé des fonctions des modules logiciels devant assurer ces fonctionnalités, est proposée une infrastructure d'intégration, inspirée de HLA et définie par une architecture fournissant des services spécifiques.*

**MOTS-CLÉS :** *aide à la décision environnementale, architecture d'intégration*

## 1. INTRODUCTION

La prise de décision environnementale peut prendre des formes très diverses, liées notamment à la nature de la décision (écologique, politique, sociale ...), au contexte décisionnel (Denzer *et al.*, 1997) (prédiction, planification à moyen/long terme, supervision et surveillance, gestion de crise d'urgence...) et aux caractéristiques du problème auquel font face les décideurs (ressources naturelles, parc naturel, incendies...). La complexité des dynamiques environnementales engendre une quantité croissante de données, d'informations et de connaissances de différentes natures et qualités, rendant la prise de décision environnementale très difficile.

En conséquence, les gestionnaires environnementaux ont un besoin croissant d'outils informatisés puissants d'aide à la décision (Janssen, 1992). De tels outils, nommés Systèmes d'Aide à la Décision Environnementale (SADE ou EDSS en anglais), se présentent comme des systèmes d'information intelligents améliorant la cohérence, la qualité, ainsi que le temps nécessaire, aux décisions environnementales (Cortès *et al.*, 2000). Ces systèmes doivent aider le(s) décideur(s) et non se substituer à lui (eux). Ils peuvent aider à comprendre et gérer un écosystème, à accumuler et analyser des informations environnementales quantitatives et qualitatives, à adapter des modèles conceptuels à la gestion locale, et sélectionner des alternatives de gestion appropriées de manière à optimiser les critères de décision (Mowrer, 1997).

Cette recherche a pour objectif de faciliter le développement de SADE, en proposant une architecture logicielle générique et réutilisable, centrée sur une infrastructure d'intégration à base de services.

Dans la deuxième section, un ensemble de projets de SADE sont étudiés en s'intéressant notamment à la nature de l'aide à la décision apportée par chacun de ces systèmes. Une analyse comparative de ces projets permet d'identifier les fonctionnalités majeures que peut présenter un SADE pour les utilisateurs décideurs.

Dans la section trois, les objectifs et descriptions fonctionnels de modules logiciels assurant ces fonctionnalités sont définis, ainsi que les solutions logicielles retenues dans le développement de SADEs existants. La grande diversité des solutions logicielles mises en œuvre pour le développement de ces systèmes, fait qu'une architecture logicielle générique qui assurerait ces fonctionnalités est difficilement envisageable. Aussi, il paraît plus réaliste de proposer une infrastructure logicielle d'intégration, permettant d'intégrer et de faire interagir diverses solutions logicielles existantes.

Dans la section quatre, l'architecture de cette infrastructure d'intégration est présentée avec son principe d'intégration, ainsi que les différents services qu'elle fournit pour assurer l'interopérabilité.

Enfin nous concluons sur les solutions logicielles présentes pour le développement de cette infrastructure d'intégration.

## 2. FONCTIONNALITES MAJEURES D'UN SADE

Dans cette section trois grands niveaux de décision environnementale, associés à l'horizon temporel, sont tout d'abord distingués : les décisions relevant de la planification stratégique, du pilotage et de la régulation. Une étude comparative de six projets de SADE significatifs permet, par la suite, d'illustrer une assistance à chacun

de ces trois niveaux. Cette étude permet également d'identifier six fonctionnalités majeures que peuvent offrir, à un niveau plus ou moins développé, des SADEs aux utilisateurs décideurs. Ces fonctionnalités doivent pouvoir couvrir la majorité des besoins des décideurs pour de tels systèmes.

### 2.1. Niveaux décisionnels concernés par les SADEs

La prise de décision environnementale s'inscrit fréquemment dans le cadre de la gestion d'un écosystème. Différents contextes décisionnels peuvent être définis en s'inspirant de la typologie des activités managériales dans l'entreprise, définie par Antony (Anthony, 1966). Trois contextes décisionnels sont ainsi distingués :

- La *planification stratégique* : activité conduisant à des décisions majeures dont la portée est très globale et les conséquences sont à long terme. Ces décisions peuvent, par exemple, engager l'avenir de tout un écosystème.
- Le *pilotage* : activité conduisant en général à des décisions dont les conséquences sont à moyen terme et dont la portée est intermédiaire entre celles de planification stratégique et de régulation.
- La *régulation* : activité qui concerne principalement des rythmes inférieurs au mois et conduit à des décisions dont les conséquences sont à court ou très court terme. Leur portée est limitée et peut ne concerner par exemple qu'une zone très précise d'un écosystème.

Un SADE, pouvant être principalement considéré comme un outil d'aide à la décision managériale en environnement, on peut ainsi situer sa zone d'intervention sur un ou plusieurs de ces niveaux.

### 2.2. Étude comparative de quelques SADEs

Afin de pouvoir dégager les fonctionnalités majeures que peuvent offrir, à un niveau plus ou moins développé, les SADEs à ses utilisateurs décideurs, six projets de SADE représentatifs (détaillés dans (Serment *et al.*, 2005)) ont été étudiés :

- Le projet *BIOMAS* (Courdier *et al.*, 2002) concernant les pratiques de gestion des effluents d'élevage et visant à diminuer les risques de pollution de l'environnement ;
- Le projet *CATCHSCAPE* (Becu *et al.*, 2003) qui s'intéresse à la gestion d'une ressource naturelle partagée ;
- Le projet *PROBIO* (Botequilha Leitão *et al.*, 2001) relatif à la prévision et la gestion de la biodiversité en zone protégée ;
- Le projet *NED* (Nute *et al.*, 2003) concernant la gestion d'écosystèmes forestiers ;
- Le projet *ALI* (Jaber, 1999) pour la prévention et la lutte contre les incendies de forêts ;
- Et le projet *WWTP* (Borrell *et al.*, 2002) traitant du contrôle et de la gestion des usines de traitement d'eaux usées.

Ces projets de SADE sont très différents notamment de part leurs objectifs et finalités. En conséquence, ses systèmes fournissent aux décideurs des types d'assistance tout aussi différents. Les niveaux décisionnels introduits précédemment permettent cependant de situer le niveau d'intervention de chacun.

Les décisions auxquelles s'intéressent les projets *BIO-MAS* et *CATCHSCAPE* relèvent principalement de la planification stratégique. Dans ces deux projets, l'objectif n'est pas de trouver une solution immédiate à un problème précis, mais de mener avec les acteurs concernés une réflexion sur un processus impliquant l'ensemble de l'écosystème. Ces projets visent à faire prendre conscience aux acteurs qu'ils font partie d'un tout, et que chacun de leurs actes a des conséquences sur le reste. Ceci peut amener à des décisions collectives ou des changements de comportement progressifs ayant des conséquences à long terme.

Les projets *PROBIO* et *NED* concernent des décisions relevant plutôt du pilotage. Ils traitent de la planification à moyen/long terme et des moyens de gestion pour maintenir les écosystèmes dans un état stable. Ils portent sur des décisions s'appliquant à des parties de l'écosystème. Les projets *ALI* et *WWTP* cherchent à apporter une aide à des décisions relevant de la régulation. Il s'agit pour ces deux projets, de surveillance et de contrôle. Ces projets concernent des problèmes (incendie, eaux usées) où les décideurs doivent réagir rapidement et dont les conséquences de ces décisions doivent être immédiates (court ou très court terme). Les solutions de ces problèmes de décision portent sur des zones précises.

### 2.3. Fonctionnalités majeures attendues des SADEs

L'étude de ces six projets de SADE, a permis d'identifier six fonctionnalités majeures, attendues par les utilisateurs de tels systèmes : la modélisation, la simulation, la visualisation, l'analyse, l'interaction avec l'utilisateur et enfin la gestion des données. Dans les paragraphes qui suivent, chacune de ces grandes fonctionnalités est tout d'abord décrite, pour ensuite illustrer leur développement dans le cadre des six projets de SADEs étudiés.

#### 2.3.1. Fonctionnalité de modélisation

Un décideur doit pouvoir caractériser le problème de décision au travers d'une multi-modélisation, et doit pouvoir prendre en compte les différents acteurs et leurs interactions. Cette fonctionnalité concerne la description des entités rentrant dans la composition des modèles écologiques, biologiques, ou sociaux économiques, sur lesquels repose l'aide à la décision. Certains de ces modèles sont des modèles de simulation. Cette description concerne le comportement de ces entités ainsi que leurs interactions dans le modèle. Plusieurs modèles peuvent être utilisés (multi-modélisation), une description de leurs interactions ou couplage est alors nécessaire.

#### 2.3.2. Fonctionnalité de simulation

Un SADE doit permettre aux utilisateurs d'élaborer différentes alternatives, de définir des scénarii et de les tester en simulant les modèles. La simulation peut être utili-

sée dans les SADEs selon différentes perspectives (Reynolds *et al.*, 1999) : la *prédiction*, la *compréhension*, l'*extrapolation* ou l'*interpolation*. La fonctionnalité de simulation attendue des SADEs concerne, tout d'abord, l'élaboration de différents scénarii et plans d'expériences<sup>1</sup>, liés aux modèles de simulation. Cette fonctionnalité doit aussi permettre le contrôle et la conduite de la simulation lors de son exécution, ainsi que l'interaction, la synchronisation, la coordination entre les différents modèles de simulation en exécution, dans des environnements de simulation pouvant être différents.

### 2.3.3. Fonctionnalité de visualisation

Pour que les résultats soient utilisables, leur visualisation doit être conviviale et paramétrable. Cette fonctionnalité doit permettre une visualisation des modèles supportant l'aide à la décision, tant dans leur structure que dans leur dynamique. Les résultats intermédiaires ou finaux d'une simulation de modèles doivent pouvoir être visualisés. L'interprétation des résultats s'en trouve facilitée, ainsi que la communication entre les différents décideurs. La visualisation doit permettre au(x) décideur(s) de se concentrer sur les entités d'un modèle, sur un modèle dans son ensemble et/ou sur les interactions entre les entités, et les interactions entre les modèles. Dans le domaine de l'aide environnementale, les modèles étant souvent spatialisés, une telle visualisation s'appuie, en général, sur un référentiel spatial prédéfini.

### 2.3.4. Fonctionnalité d'analyse

Une fois simulées, les différentes alternatives doivent pouvoir être évaluées. Cette fonctionnalité propose une assistance à l'analyse de résultats, notamment de simulation, ainsi qu'une aide à la comparaison de différentes alternatives simulées. Cette fonctionnalité doit permettre à l'utilisateur de faire aussi bien une analyse approfondie des résultats d'une simulation ou d'un plan d'expérience, qu'une analyse comparative de simulations basée sur des critères de décision. L'utilisateur peut, par exemple, s'appuyer sur des méthodes statistiques ou multicritères.

## 2.4. Fonctionnalités auxiliaires

Les fonctionnalités auxiliaires sont des fonctionnalités transversales aux fonctionnalités majeures de modélisation, simulation, visualisation et analyse, permettant d'intégrer l'utilisateur et de gérer les données.

### 2.4.1. Fonctionnalité d'intégration de l'utilisateur

Un SADE étant un système d'aide à la décision, l'interaction entre le(s) décideur(s) et le système doit être appréhendée en tant que fonctionnalité majeure du SADE. Cette fonctionnalité doit supporter cette interaction utilisateur – système dans le cadre de chacune des fonctionnalités majeures déjà introduites. Cette intégration de l'utilisateur peut aller jusqu'à sa participation au processus de modélisation (jeu de rôle, approche participative...), ou dans l'exécution d'une simulation. Cette

fonctionnalité doit permettre un paramétrage de la simulation, la définition de plan d'expérience et de scénario. Elle peut aussi lui permettre d'intervenir dans le pilotage de la simulation. Les simulations de type « steering » (Campos, 2000) offrent la possibilité aux utilisateurs d'interagir avec le modèle durant son exécution, pour pouvoir modifier ses paramètres dynamiquement. Relativement à la visualisation, cette fonctionnalité d'interaction doit permettre aux décideurs de choisir quoi et comment l'afficher. L'utilisateur doit également pouvoir définir les résultats de simulation(s) à analyser et de quelle manière.

### 2.4.2. Fonctionnalité de gestion des données

Cette fonctionnalité concerne le traitement des données, tant exogènes (externes) qu'endogènes (internes), associées à la réalisation des diverses autres fonctionnalités. Elle doit ainsi permettre de stocker les données, notamment spatiales et temporelles, relatives aux modèles et de les gérer pour leur utilisation notamment dans la simulation. Elle doit également prendre en charge les données relatives aux résultats de simulation intermédiaires et finaux. Cette fonctionnalité est en lien étroit avec la fonctionnalité de visualisation devant permettre l'affichage, dans une certaine forme, de certaines données. Enfin, elle est en interaction avec la fonctionnalité d'analyse utilisant les données des résultats de simulation pour les exploiter mais également pour stocker les résultats de ces comparaisons et analyses. Ces deux fonctionnalités peuvent éventuellement interagir pour la récupération de données issues du réel dans l'optique d'une validation de modèle.

Dans les six projets de SADE étudiés, l'importance du développement de ces fonctionnalités dépend du type de décision concerné. La figure 2 met en avant l'importance des fonctionnalités de modélisation (M), de simulation (S), de visualisation (V) et d'analyse (A) selon le niveau d'aide à la décision concerné (planification stratégique, pilotage et régulation).

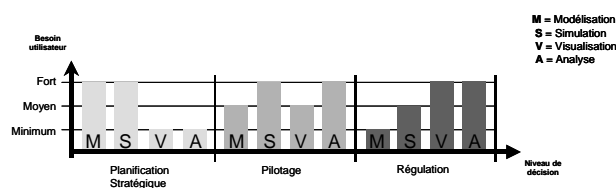


Figure 1. Importance des fonctionnalités et niveaux de décisions

Ainsi, les projets *BIOMAS* et *CATCHSCAPE*, qui relèvent principalement de la *planification stratégique*, sont basés sur la compréhension des modèles et de leurs interactions. L'outil doit permettre aux utilisateurs (non informaticiens) de modifier eux-mêmes les modèles et d'intervenir en cours de simulation sur le système pour mieux appréhender ses réactions.

Ces besoins sont utiles, à un degré moindre, pour une aide à la décision relevant du *pilotage*, comme pour les projets *PROBIO* et *NED*, où l'utilisateur doit surtout

<sup>1</sup> Nous utilisons les plans d'expérience comme définis dans (Hill, 1997) pour valider et tester la robustesse d'un modèle, mais également, dans notre contexte d'aide à la décision, pour vérifier qu'il soit une solution possible et pour évaluer cette solution.

pouvoir élaborer et évaluer un grand nombre d'alternatives. Ce sont les modules de simulation et d'analyse avec lesquels l'utilisateur interagit le plus.

Enfin, pour les projets *ALI* et *WWTP* les décisions relèvent de la *régulation*. Les modèles ne peuvent pas, ou très peu, être modifiés, mais les résultats de simulation doivent être évalués de manière rapide et précise. Le module d'analyse doit fournir une analyse précise des résultats, et le module de visualisation doit rendre ces résultats d'analyse rapidement compréhensible et permettre de suivre la simulation avec un niveau de détails plus fin.

### 3. FONCTIONS DES MODULES LOGICIELS COMPOSANT UN SADE

La partie précédente, correspondant à l'expression des besoins, montrait qu'en général, un SADE devait proposer six grandes fonctionnalités. Cette section décrit les objectifs, fonctions et interactions de différents modules logiciels génériques composant un SADE et assurant ces fonctionnalités.

Chacune des six fonctionnalités définies dans la section précédente peut être réalisée par un module logiciel spécifique. L'architecture générique d'un SADE repose ainsi sur six modules logiciels en interaction : un *module de modélisation*, un *module de simulation*, un *module de visualisation*, un *module d'analyse*, un *module d'intégration utilisateur* et un *module de gestion des données*. Un module se définit comme un ensemble de fonctions ayant un nom par lequel les autres parties du système peuvent l'invoquer. Ces modules manipulent un certain nombre d'informations pouvant être échangées avec les autres modules.

Les sous-sections suivantes présentent l'ensemble des fonctions devant être assurées par chacun des modules, sans tenir compte des différences par niveaux de gestion. Les fonctions ne seront pas utilisées de la même façon selon le type de décision. Cette décomposition des modules fait ressortir les interactions entre les modules.

Il existe une grande diversité d'architectures logicielles pouvant être retenues pour chacun de ces modules, notamment dans la manière d'utiliser les technologies objet et agent. Toujours dans le but de caractériser les besoins d'un SADE générique, des choix possibles pour chacun des modules sont énumérés.

#### 3.1. Module de modélisation

L'objectif de ce module est la construction des modèles. Les modèles abordés ici sont les modèles exécutables qui seront ensuite utilisés par le simulateur. Ces modèles peuvent être des modèles multi-agents, des modèles objets, des automates cellulaires ou autres solutions d'opérationnalisation de modèles individu-centrés. Ce module permet la description des entités, de leur comportement et de leurs interactions. Il doit également permettre la description des interactions (couplage) entre

plusieurs modèles ou reposer sur un méta modèle comme dans le projet MIMOSA (Müller, 2004).

##### 3.1.1. Description fonctionnelle

Sont détaillées ici, les informations traitées et les fonctions et sous-fonctions de ce module de modélisation permettant la construction de modèles.

**Informations traitées :** *description de modèles, description de couplages de modèles.*

##### Principales fonctions à assurer :

*Définir/Modifier un modèle :* Choix du type de modèle (paradigme), choisir un modèle existant, Définir/Modifier un type d'entité, Définir/Modifier l'organisation des entités, Définir/Modifier le nombre d'entités, Définir/Modifier les interactions entre les différentes entités, Proposer des méthodes de construction de modèles.

Par exemple pour définir un modèle social, l'utilisateur peut choisir d'utiliser le paradigme Agent, définir un type d'Agent en spécifiant ses connaissances et comportements, définir un groupe (au sens AGR (Ferber *et al.*, 2003)) localisant les négociations entre différents types d'agent et définir les protocoles d'interaction au sein de ce groupe. Le choix d'une méthode de construction peut par exemple consister à considérer que les entités Maille d'un modèle correspondent à la somme des cellules d'un même type de culture sur un même bassin versant (Franchesquin, 2001).

*Définir/Modifier les interactions entre les modèles de simulation :* Sélection des modèles, Choix du type d'interaction, Spécifier les interactions. Par exemple, définir qu'un modèle social et un modèle hydraulique vont interagir, et que le couplage se fait par les données d'irrigation, qui sont des résultats des décisions du modèle social et des paramètres d'entrée du modèle hydraulique.

##### 3.1.2. Solutions logicielles existantes

Dans ce module, différentes approches sont utilisées pour la modélisation : automate cellulaire, objet, agents réactifs, agents hybrides, agents cognitifs. Les agents cognitifs sont utilisés pour la représentation de processus sociaux (négociation, coopération...) ou décisionnels. Ils peuvent également servir à l'approche participative (Bousquet et Le Page, 2004) de par leur capacité à représenter les comportements des acteurs sociaux. Les agents réactifs permettent la représentation de phénomènes émergents et sont particulièrement adaptés à la représentation de comportements d'individus en forte interaction. Des agents hybrides peuvent également être utilisés (Andriamasinoro, 2003) pour permettre la prise en compte du comportement réactif d'agents cognitifs. L'approche Objet est généralement utilisée pour la représentation d'entités au comportement plus simple ou pour des modèles analytiques. Ce module doit également offrir la possibilité à l'utilisateur de s'appuyer sur des formalismes reconnus tels que UML et AUML ou encore DEVS.

### 3.2. Module de simulation

Ce module est central pour l'aide à la décision, c'est lui qui permet de tester les différentes solutions possibles. Il doit pouvoir assurer la simulation des différents modèles. Il permet de tester différents scénarii et plans d'expériences sur ces modèles. Il doit assurer le contrôle et la conduite de la simulation ainsi que la coordination entre les différents modèles.

#### 3.2.1. Description fonctionnelle

Sont détaillées ici, les informations traitées et les fonctions et sous-fonctions de ce module de simulation.

**Informations traitées :** *Descriptions de modèles, Descriptions de couplages, Facteurs d'initialisation du modèle, Scénario de simulation, Résultats de simulation*

#### Principales fonctions assurées :

*Paramétrer le simulateur :* Choisir le pas de temps, choisir la durée de la simulation, paramétrer le générateur de nombre aléatoire, ...

Par exemple, choisir entre un pas de temps journalier ou mensuel selon le type de décisions, définir une simulation sur quelques mois pour des décisions de type régulation ou plusieurs années pour des décisions de types planification stratégique.

*Définir/Modifier un scénario de simulation :* Définir/Modifier les facteurs d'initialisation, Définir/Modifier un calendrier<sup>2</sup>, Définir/Modifier la configuration<sup>3</sup> de modèle.

Par exemple, choisir la surface pour chaque type de culture des agents exploitation agricole, définir les événements externes arrivant à chaque pas de temps (pluie, évapotranspiration...), Supprimer un type de culture pour un agent exploitation.

*Définir/Modifier un plan d'expérience :* Choisir/définir/Modifier les modèles communs ou le calendrier commun.

Le(s) décideur(s) peut par exemple vouloir tester différentes configurations de modèles sur une même durée ainsi que des événements externes identiques pour pouvoir les comparer par la suite, ou tester une configuration de modèle de nombreuse fois en faisant varier les événements externes afin de l'évaluer.

*Conduire la simulation :* Lancer ou relancer un scénario de simulation, Mettre en pause, Arrêter la simulation, Revenir en arrière.

*Intervenir en cours de simulation :* Modifier, ajouter des données au calendrier, Modifier un paramètre, Modifier (ou ajuster) les modèles.

*Définir les résultats intermédiaire et finaux à enregistrer*

<sup>2</sup> Nous définissons un calendrier comme l'ensemble des événements externes ayant une date. C'est à dire devant être déclenchés à un moment t donné par le simulateur.

<sup>3</sup> Une configuration de modèle concerne aussi bien le niveau statique (nombre d'entités, attributs, constantes) que dynamique (méthodes, interactions)

#### 3.2.2. Solutions logicielles existantes

Différents environnements de simulation assurant la simulation des modèles peuvent faire partie de ce module. Les concepts objets et agents sont également présents ici, bien qu'ils ne correspondent pas forcément à l'approche utilisée pour la modélisation. Ainsi un modèle multi-agents peut être simulé dans un simulateur à base d'objet. Ceci est peut-être dû au fait qu'il n'y ait pas de langage de programmation agent encore réellement utilisable. Le simulateur lui-même peut avoir été développé selon une approche agent comme la plate-forme MadKit (Ferber *et al.*, 2000) ou Cormas (Becu *et al.*, 2003).

### 3.3 Module de visualisation

Il permet la visualisation des comportements des modèles et des résultats de simulation. Ce module permet de suivre l'évolution de la simulation en affichant les résultats intermédiaires et finaux de manière graphique. L'interprétation des résultats s'en trouve améliorée, ainsi que la communication entre les différents experts. A travers cette visualisation, les décideurs doivent pouvoir se concentrer sur les entités d'un modèle, sur un modèle dans son ensemble et/ou sur les interactions entre les entités et les interactions entre les modèles.

#### 3.3.1. Description fonctionnelle

Le module de visualisation est fortement lié et dépendant des autres modules. Sa décomposition en fonctions met en avant les collaborations avec les autres modules. Les interactions entre les autres modules et ce module permettent d'améliorer la communication avec l'utilisateur. Les informations traitées et les principales fonctions assurées par ce module sont synthétisées dans le tableau 2.

#### 3.3.2. Solutions logicielles existantes

Ce module peut-être composé d'outils de visualisation graphique adaptés aux données géographiques, comme par exemple Geotools (Geotools, 2006). La programmation Orientée Objet, avancée dans ce domaine, est la plus souvent utilisée, bien que l'utilisation d'agents interface soit de plus en plus fréquente. Un des avantages de l'approche agent est que la visualisation peut directement être liée aux entités simulées (Campos, 2000), permettant un meilleur suivi de la simulation. Les agents des modèles peuvent également fournir eux-mêmes une interprétation de l'état du système (Servat *et al.*, 1998).

### 3.4. Module d'analyse

Ce module offre la possibilité d'analyser des résultats et de comparer différentes alternatives simulées. Il doit permettre à l'utilisateur de faire aussi bien une analyse approfondie des résultats d'une simulation ou d'un plan d'expérience, qu'une analyse comparative de simulations basées sur des critères de décision. Pour cela, l'utilisateur peut s'appuyer sur des méthodes statistiques ou multicritères.

#### 3.4.1. Description fonctionnelle

Le module d'analyse est décomposable en fonctions correspondant à des choix pouvant être faits par les gestionnaires. Ce module est principalement en interaction avec

les modules de gestion des données et de visualisation. Les informations traitées et les principales fonctions assurées par ce module sont synthétisées dans le tableau 2.

### 3.4.2. Solutions logicielles existantes

Des agents cognitifs peuvent être utilisés pour l'analyse des résultats. Il peut exister deux niveaux de décision : des décisions prises par les agents des modèles, et des décisions prises après la simulation par des agents logiciels ayant des connaissances du domaine. Les agents peuvent avoir un modèle de décision se raccordant le plus souvent à l'approche multicritère. Des agents interface peuvent également être utilisés pour la comparaison des résultats de simulations différentes.

### 3.5. Module d'intégration de l'utilisateur

Le(s) décideur(s) sont considérés, dans ce travail, comme faisant partie du système. Ce module permet à l'utilisateur d'accéder aux différentes fonctions assurées par le SADE. Ces fonctions étant assurées par différents modules, il permet l'interaction entre l'utilisateur et ces modules. L'utilisateur accède aux fonctions des différents modules, sans avoir à savoir quel module assure cette fonction. Il permet également à l'utilisateur d'accéder à des fonctions internes au SADE.

#### 3.5.1. Description fonctionnelle

Les fonctions à assurer par ce module permettent d'accéder aux fonctions des autres modules. Il doit permettre la participation des utilisateurs au processus de modélisation, au paramétrage de la simulation, de piloter et d'intervenir en cours de simulation, paramétrer la visualisation et définir les résultats de simulation(s) à analyser et comment. Les informations traitées et les principales fonctions assurées par ce module sont synthétisées dans le tableau 2.

#### 3.5.2. Solutions logicielles existantes

Des agents interface et des agents logiciels peuvent être utilisés pour permettre une meilleure intégration de l'utilisateur au sein du système, fournissant des interfaces personnalisés, adaptables et pouvant offrir une harmonisation entre les différents outils utilisés. Les agents des modèles de simulation peuvent également permettre son intégration au sein des modèles.

### 3.6. Module de gestion des données

Ce module prend en charge le traitement des données. Tout comme le module d'intégration de l'utilisateur, il s'agit d'un module transversal sur lequel peut reposer l'interopérabilité des modules. Il doit permettre de stocker les données, notamment spatiales et temporelles, relatives aux modèles et de les gérer pour leur utilisation dans la simulation et la coopération entre les différents modules. Il doit également prendre en charge les résultats intermédiaires et finaux. Il est en lien étroit avec le module de visualisation devant permettre l'affichage dans une certaine forme de certaines données. Enfin, il doit pouvoir interagir avec le module d'analyse utilisant les données des résultats de simulation pour les exploiter, mais également pour stocker les résultats de ces

comparaisons et analyses. Ces deux modules peuvent éventuellement interagir pour la récupération de données réelles du système à modéliser, dans l'optique d'une validation de modèle.

#### 3.6.1. Description fonctionnelle

Ce module doit assurer des fonctions permettant de manipuler les données. Les informations traitées et les principales fonctions assurées par ce module sont synthétisées dans le tableau 2.

#### 3.6.2. Solutions logicielles existantes

Des agents logiciels peuvent également être utilisés dans ce module. Ces agents permettent de superviser les systèmes de gestion de base de données et SIG facilitant le couplage (Maillé et Espinasse, 2005) et l'interopérabilité modulaire. De nombreuses solutions logicielles existent pour la gestion de base de données ou systèmes d'information géographique.

Le tableau 2 synthétise les informations traitées et les fonctions (non détaillées) assurées par chacun des modules logiciels.

	Informations traitées	Principales Fonctions assurées
<b>Modélisation</b>	Description de modèles Description de couplages	Définir/Modifier un modèle Définir/Modifier les interactions entre les modèles de simulation
<b>Simulation</b>	Description de modèles Description de couplages Facteurs d'initialisation du modèle Scénario de simulation Résultats de simulation	Paramétrer le simulateur Conduire la simulation Intervenir en cours de simulation Définir les résultats intermédiaire et finaux à enregistrer Définir/Modifier un scénario de simulation Définir/Modifier un plan d'expérience
<b>Visualisation</b>	Données géographiques Description de modèles Résultats de simulation Résultats d'analyse Paramètres de visualisation	Afficher modèles Paramétrer la visualisation de la simulation Afficher le suivi de la simulation Paramétrer la visualisation de l'analyse Afficher les résultats de l'analyse
<b>Analyse</b>	Résultats de simulation Données de « repère » Résultats d'analyse	Analyser les résultats de simulation Analyser les résultats d'un plan d'expérience Analyse pour visualisation
<b>Utilisateur</b>		Accéder aux fonctions de modélisation Accéder aux fonctions de simulation Accéder aux fonctions de visualisation Accéder aux fonctions d'analyse Accéder aux fonctions de gestion de données
<b>Données</b>	Descriptions de modèles Descriptions de couplages Paramètres d'initialisation Scénario de simulation Résultats de simulation Données géographiques Résultats d'analyse Paramètres de visualisation Données de « repère »	Enregistrer une donnée Modifier une donnée Récupérer une donnée Supprimer une donnée S'abonner à une donnée

Tableau 2. Information et fonction des modules

Ce tableau et le détail des modules de modélisation et simulation font ressortir, notamment au niveau des informations traitées, les connexions entre les modules.

### 3.7. Interactions entre modules

Les principales interactions entre modules logiciels sont décrites dans le tableau 3, ci-dessous.

Le module de Modélisation (M) produit et fournit aux modules interagissant avec lui la description de modèles et de leurs couplages. Le module de simulation (S) utilise ces informations pour pouvoir exécuter ces modèles. Il produit principalement des résultats de simulation uti-

lisés par d'autres modules. Le module de visualisation (V) reçoit des informations de l'ensemble des autres modules du SADE et permet leur visualisation. Le module d'analyse (A) produit des résultats d'analyse à partir de résultats de simulation. Ces résultats sont fournis aux modules de visualisation et de gestion des données (D). Le module d'intégration de l'utilisateur (U) permet à l'utilisateur de faire appel aux différents modules. Enfin, le module de gestion de données reçoit et fournit des informations à l'ensemble des modules du système.

↗	M	S	V	A	U	D
M		X	X		X	X
S			X	X	X	X
V					X	X
A			X		X	X
U	X	X	X	X		X
D	X	X	X	X	X	

Tableau 3. Matrice d'interaction

Cette matrice fait ressortir l'aspect transversal des modules assurant les fonctionnalités auxiliaires. En effet, les modules d'intégration de l'utilisateur et de gestion des données sont en interaction avec l'ensemble des autres modules. Ce tableau ne représente pas les possibilités d'interactions au sein d'un même module, comme par exemple dans le cas où plusieurs modèles de simulation sont couplés.

La figure 2 représente un diagramme de séquence permettant d'illustrer ces interactions en tenant compte de la dimension temporelle et faisant ressortir le besoin d'interopérabilité entre les modules. L'interopérabilité est considérée dans ce travail comme la possibilité, pour ces modules, de pouvoir faire appel aux services d'un autre module et d'échanger des informations.

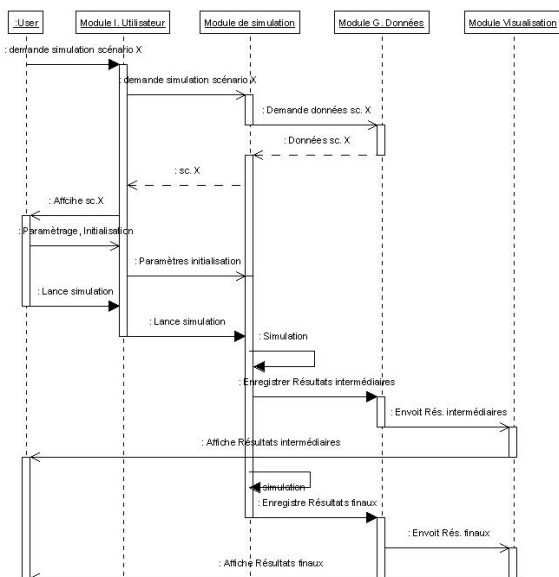


Figure 2. Diagramme de séquence « exécution d'un scénario de simulation défini ultérieurement »

Pour exécuter correctement une fonction, un module peut avoir besoin de faire appel aux fonctions ou données d'un autre module. Comme l'illustre le diagramme

de la figure 2, pour réaliser l'exécution d'un scénario, le module utilisateur appelle le module de simulation. Le module de simulation qui a besoin d'informations appelle ensuite le module de donnée, en reçoit du module utilisateur, et exécute des fonctions internes lui permettant de réaliser la simulation. Enfin, le module de simulation va fournir les résultats aux modules de gestion des données et de visualisation. L'architecture du SADE doit permettre cette interopérabilité.

### 3.8. D'une architecture générique à une architecture d'intégration

Les modules logiciels devant rentrer dans la composition d'un SADE, et permettant de répondre aux besoins des utilisateurs, ont été identifiés. Les objectifs et fonctions de chacun de ces modules logiciels ont été définis. Et ont également été évoquées les solutions logicielles retenues dans le développement de SADEs existants.

La grande diversité de ces solutions logicielles mises en œuvre pour le développement de ces systèmes, fait qu'une architecture logicielle générique qui assurerait ces fonctionnalités est difficilement envisageable. Aussi, il paraît plus réaliste de proposer une infrastructure logicielle d'intégration permettant d'intégrer et de faire interagir diverses solutions logicielles existantes.

Une telle infrastructure doit permettre d'intégrer des solutions existantes pour chaque module au sein d'un même système (Figure 3) et de les faire interagir. Une telle infrastructure permettrait ainsi d'offrir toutes les fonctionnalités attendues et ne serait pas remise en cause à chaque nouveau type de décision. Elle ne serait pas dédiée à un seul SADE, mais faciliterait le développement de différents SADEs.

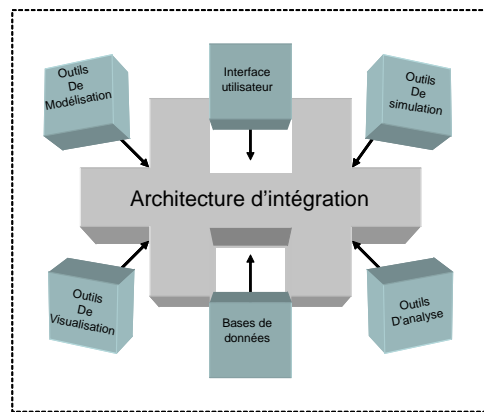


Figure 3. Intégration des modules

L'objectif de ce travail n'est pas de déterminer quelles solutions logicielles utiliser pour chacun des modules, mais de permettre au concepteur d'utiliser la solution logicielle de son choix et de pouvoir l'intégrer à la plateforme. L'architecture doit permettre la conception d'un SADE constitué des solutions logicielles les mieux adaptées au problème à traiter. L'architecture doit faciliter l'intégration de ces systèmes, permettre à l'utilisateur de les manipuler et les faire interagir.

## 4. UNE INFRASTRUCTURE D'INTÉGRATION POUR LA CONCEPTION DE SADE

Dans cette section, l'infrastructure d'intégration proposée pour le développement de SADE est présentée. Dans un premier temps est introduite l'architecture de cette infrastructure d'intégration et les différents éléments la composant, puis sont présentés son principe d'intégration ainsi que les différents services fournis par chacun de ces composants pour assurer l'interopérabilité.

### 4.1. Définition de l'architecture de l'infrastructure

L'architecture de l'infrastructure d'intégration proposée s'inspire du standard HLA (DMSO, 1996). Ce standard de la simulation distribuée est une architecture logicielle et une spécification d'interface. Développée sous le contrôle du DMSO<sup>4</sup>, HLA assure l'interopérabilité entre différents simulateurs. L'infrastructure décrite ici doit permettre l'interopérabilité entre les modules dits « fonctionnels », modules réalisant les fonctionnalités majeures des SADES définies précédemment (modélisation, simulation, visualisation et analyse). Le module de gestion des données fait partie intégrante de l'infrastructure.

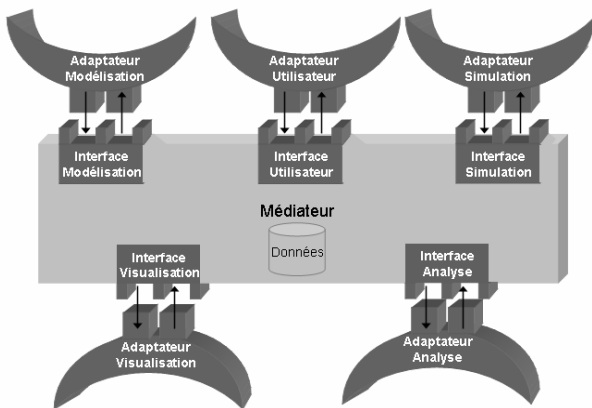


Figure 4. Architecture d'intégration pour le développement de SADE

L'architecture de l'infrastructure d'intégration (Figure 4) est composée des éléments suivants :

- Des *adaptateurs* : Chaque module fonctionnel à intégrer est pris en charge (ou encapsulé) par un adaptateur (wrapper en anglais) assurant son intégration au système.
- Des *interfaces* : Les interfaces permettent de faire le lien entre ces adaptateurs et le médiateur.
- Un *médiateur* : c'est sur lui que repose l'interopérabilité, il va permettre aux différents modules à intégrer d'interagir via les interfaces.

Ces composants de l'infrastructure communiquent entre eux pour permettre l'intégration et l'interopérabilité entre les différents modules à intégrer.

### 4.2. Principe de l'intégration

L'infrastructure proposée doit permettre aux différents modules fonctionnels d'être intégrés au système et d'interagir. L'intégration repose sur l'acheminement d'un appel de fonction d'un module fonctionnel vers le médiateur et inversement. L'interopérabilité repose sur le bon acheminement d'un appel émanant d'un module fonctionnel donné vers le module fonctionnel capable d'assurer cette fonction.

L'infrastructure doit gérer à la fois l'intégration logicielle, prise en charge par l'adaptateur, et l'interopérabilité sémantique, par le biais des interfaces.

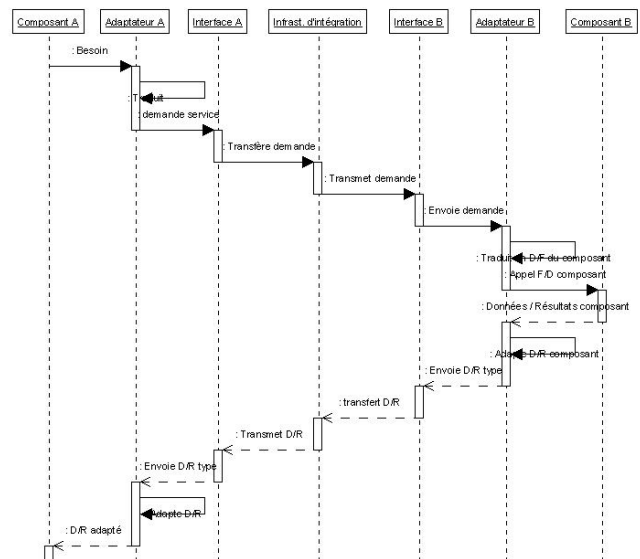


Figure 5. Principe de l'interaction

L'interface doit être capable de représenter le module fonctionnel auprès des autres interfaces et du médiateur notamment en relayant des besoins de données ou de services. Ce dialogue implique un langage commun aux interfaces et au médiateur permettant d'exprimer ces besoins.

L'interface doit également transférer à son adaptateur une demande de données ou de fonctions, récupérer le résultat de cette requête et acheminer ce résultat vers l'interface qui l'a sollicité ou le médiateur. L'adaptateur et l'interface doivent également être capable de communiquer en s'appuyant sur un langage et une ontologie communs, indépendants des solutions logicielles adoptées pour le module fonctionnel, mais spécifiques à chaque couple adaptateur-interface. L'adaptateur reçoit dans ce même langage la requête transmise par l'interface et la traduit en appel (niveau API) au composant qu'il « encapsule ».

L'interopérabilité est assurée par le médiateur. Il coordonne le dialogue entre les interfaces, mais assure également les fonctions de synchronisation entre les modules fonctionnels (gestion du temps, accès à la base de donnée etc.). Cette architecture étant indépendante du niveau d'aide souhaité, l'utilisateur doit pouvoir utiliser les modules différemment en fonction du type de déci-

<sup>4</sup> Defense Modeling and Simulation Office



sion. Pour cela, le médiateur doit contrôler et faire varier l'enchaînement des échanges en fonction du niveau de gestion.

Le problème ainsi posé se complique dès lors qu'un SADE peut être composé de plusieurs instances d'un même type de module fonctionnel (par exemple, deux modules de simulation couplés), voire s'appuyant sur des solutions logicielles différentes (par exemple, un simulateur agent et un simulateur DEVS). L'interopérabilité logicielle étant assurée par l'adaptateur et l'interopérabilité sémantique par l'interface, il faut définir autant d'adaptateurs que de modules fonctionnels implémentés, et autant d'interfaces que de type de module (interface modélisation, interface simulation etc... soit cinq en tout). Le dialogue entre un couple adaptateur-interface implique par ailleurs de définir une ontologie commune (ontologie simulation, ontologie modélisation etc...).

#### 4.3. Les services assurés par l'infrastructure

Il existe deux niveaux de services : des services issus des besoins d'aide à la décision et des services issus des besoins liés à l'intégration et l'interaction des modules fonctionnels logiciels. Les services liés aux besoins utilisateur sont assurés par les modules décrits dans la section précédente. Les services étudiés dans cette section sont les services proposés par l'infrastructure pour assurer l'intégration et l'interopérabilité des modules fonctionnels à intégrer. L'infrastructure d'intégration est le support offrant ces différents services, chacun des éléments devant assurer ses propres services.

*Services assurés par les adaptateurs :*

- Appel à une donnée ou fonction du module fonctionnel qu'il enveloppe.
- Communication avec l'interface correspondante.
- Traduction des demandes émanant de l'interface en appel de donnée ou fonction du module fonctionnel.
- Traduction des besoins des modules fonctionnels en demande à envoyer à l'interface.

*Services assurés par les interfaces :*

- Communication avec l'adaptateur correspondant.
- Communication avec le médiateur.
- Transfert d'une demande émanant de l'adaptateur au médiateur.
- Transfert d'une demande émanant du médiateur à l'adaptateur.
- Appel aux services du médiateur.

*Services assurés par le médiateur :*

- Communication avec les interfaces.
- Transmission d'une demande d'une interface vers la bonne interface.
- Accès aux données « communes ».
- Contrôle les flux de données.
- Contrôler les demandes
- Synchronisation
- Cohésion

Pour pouvoir assurer tous ces services, l'infrastructure doit permettre à ses différents éléments de communiquer entre eux.

## 5. CONCLUSION

Le besoin de généricité, ou du moins d'homogénéité, des outils de conception de SADE apparaît nécessaire, il restait à évaluer la possibilité de produire un SADE générique. À partir de l'étude de six projets de SADE, nous avons alors cherché à établir les grandes fonctionnalités nécessaires, les caractériser sous forme de modules logiciels et enfin établir les solutions logicielles pouvant les assurer. Devant la difficulté de la tâche (enjeux et niveaux décisionnels, paradigme de modélisation différents, solutions logicielles variées), il est apparu plus réaliste de proposer une infrastructure logicielle d'intégration, permettant d'intégrer et de faire agir ensemble diverses solutions logicielles existantes. L'architecture de cette infrastructure d'intégration proposée repose sur trois composants : l'adaptateur, l'interface et le médiateur. Les différents services fournis par chacun de ces composants pour assurer l'interopérabilité ont ensuite été définis.

Actuellement nous poursuivons la spécification fonctionnelle de l'infrastructure d'intégration. Compte tenu de la nature des services que doivent assurer chaque composant de l'infrastructure, une approche de spécification orientée agents a été retenue. En effet, les adaptateurs, les interfaces et le médiateur sont des composants relativement autonomes et en étroite interaction. Ces interactions doivent s'adapter selon le contexte d'intégration, par exemple en fonction de la nature des modules de modélisation ou de simulation, mais aussi selon le scénario d'utilisation en cours. La spécification de telles interactions nécessite un langage d'interaction générique. Aussi cette infrastructure est appréhendée comme un système multi-agents, chaque agent étant un composant de l'infrastructure. Au delà de cette spécification fonctionnelle, l'approche agent peut se révéler aussi pertinente dans une phase de conception logicielle. D'autres approches nous semblent aussi devoir être considérées comme celles de la programmation à base de services et des Web Services.

Cette infrastructure d'intégration spécifiée, puis réalisée, sera validée par le développement d'un SADE relatif à la gestion hydraulique de la Camargue. Des travaux ayant déjà été développés, à ce sujet, au sein de notre laboratoire, dans le cadre du projet SimFonHyC (Franchesquin, 2001) (Espinasse et Franchesquin, 2005), l'objectif est d'étendre les capacités du SADE tout en réutilisant et intégrant les modules logiciels déjà développés.

## REFERENCES

- Andriamasinoro F., 2003. *Proposition d'un modèle d'agents hybrides basé sur la motivation naturelle*. Thèse de Doctorat, IREMI, Université de La Réunion, France.
- Anthony R.N., 1966. *Planning and Control Systems : a Framework for Analysis*. Cambridge, Mass, Harvard University Press.
- Becu N., Perez P., Walker A. Barreteau O., and Le Page C., 2003. Agent based simulation of a small catchment water management in northern Thailand, Description of the CATCHSCAPE model. *Ecological Modelling* 170, Elsevier, p.319-331. <http://cormas.cirad.fr/fr/applica/catchscape.htm>
- Borrell F. Riaño D., Sánchez-Marrè M. and Rodriguez-Roda I., 2002. Implementation of a Multiagent Prototype for WWTP Management. *IEMSS, Integrated Assessment and Decision Support System*, Lugano, Suisse.
- Botequilha Leitão A., Grueau C., and al., 2001. Decision Support System for Planning and Management of Biodiversity in Protected Areas, The research project PROBIO. *Proceedings of the international Workshop on Geo-Spatial Knowledge Processing for Natural Resource Management*. Varese, Italy, p. 145-151. <http://alfa.ist.utl.pt/~cvrm/projects/probio>
- Bousquet F. and Le Page C., 2004. Multi-agent simulations and ecosystem management : a review. *Ecological Modelling* 176, p. 313-332,
- Campos A.M.C., 2000. *Une architecture logicielle pour le développement de simulations visuelles et interactives individus-centrées : application à la simulation d'écosystèmes et à la simulation sur le Web*. Thèse de Doctorat, Université Blaise Pascal – Clermont II, France.
- Cortès U., Sanchez-Marrè M. and Ceccaroni L., 2000. Artificial Intelligence and Environmental Decision Support Systems. *Applied Intelligence* 13 (1), p.5-6.
- Courdier R., Guerrin F., Andriamasinoro F.H., and Paillet J.M., 2002. Agent-based simulation of complex systems : application to collective management of animal wastes. *Journal of Artificial Societies and Social Simulation*, vol. 5, no.3. <http://www2.univ-reunion.fr/~mas2/>
- DMSO. High Level Architecture. <https://www.dms0.mil/public/transition/hla/>
- Denzer D., Swayne A. and Schimak G., 1997. *Environmental Software Systems*. Chapman & Hall, London, England.
- Espinasse B. and Franchesquin N., 2005. Multiagent Modelling and Simulation of Hydraulic Management of the Camargue. *Simulation*, Vol. 81, Issue 3, p.201-221.
- Ferber J., Gutknecht O. and Michel F., 2000. *MadKit : une plate-forme multi-agent générique*. Rapport de recherche n°R.R.LIRM 00061, LIRM.
- Ferber J., Gutknecht O. and Michel F., 2003. Agent/Groupe/Roles : Simulation with organization. *4th workshop on agent-based simulation*, SCS, Montpellier, France.
- Franchesquin N., 2001. *Modélisation et simulation multi-agents d'écosystèmes anthropisés : une application à la gestion hydraulique en Grande Camargue*. Thèse de Doctorat, Université Aix-Marseille III, France.
- GeoTools. <http://www.geotools.org/>
- Jaber A., 1999. *Un système d'agents logiciels intelligents pour favoriser la coopération entre des systèmes d'aide à la décision dédiés à la prévention des risques naturels*. Thèse de Doctorat, Ecole des Mines de Paris, France.
- Janssen R., 1992. *Multiobjective Decision Support for Environmental Management*. Kluwer Academic Publishers.
- Maillé E. and Espinasse B., 2005. Du couplage de systèmes à l'intégration spatiotemporelle dans les systèmes d'aide à la décision spatiale. *CABM-HEMA-SMAGET05*, Bourg-saint –Maurice, France.
- Mowrer, H.T., 1997. *Decision support systems for ecosystem management: an evaluation of existing systems*, General Technical Report RM-GTR-296.
- Müller, J. P., 2004. *MIMOSA : représentation des connaissances et simulation*. <http://lil.univ-littoral.fr/Mimosa>
- Nute D., Potter, W.D., Maier F., Wang J., Twery M., Rauscher H.M., Knopp P., Thomas S., Dass M., Uchiyama H. and Glende A., 2003. NED-2 : an agent-based decision support system for forest ecosystem management. *Environmental Modelling & Software*, Elsevier. <http://www.fs.fed.us/ne/burlington/ned/>
- Reynolds, K., Bjork, J., Rienmann, H.R., Schmoldt, D., Payne, J., King, S., Moeur, M., DeCola, L. Twery, M. Cunningham, P. and Lessard, G., 1999. Decision Support For Ecosystem Management. *Ecological Stewardship: A Common Reference for Ecosystem Management*, Elsevier Science Ltd., p. 687-722.
- Serment J., Espinasse B., and Tranvouez E., 2005. Contribution à une architecture logicielle générique pour le développement de système d'aide à la décision environnementale : Illustration à travers le cas Camarguais. *CABM-HEMA-SMAGET05*, Bourg-saint –Maurice, France.
- Servat, D., Perrier E., Treuil J-P. and Drogoul A., 1998. When Agents Emerge from Agents: Introducing Multi-Scale Viewpoints in Multi-Agent Simulations. *Proceedings of MABS'98*, 183--198, LNAI n° 1534, Springer-Verlag, Berlin, Germany.