AeDEn: un cadre général pour une distribution adaptative et dynamique des applications en environnements mobiles

F. Le Mouël* **et F. André†**

**IRISA/INRIA*
†IRISA/UNIVERSITÉ DE RENNES 1
Campus de Beaulieu
35042 Rennes Cedex, FRANCE
email: {flemouel,fandre}@irisa.fr

RÉSUMÉ— L'informatique nomade est un secteur en pleine évolution aussi bien au niveau des terminaux mobiles (ordinateurs portables, assistants personnels, téléphones portables, etc), qu'au niveau de l'infrastructure et des protocoles des réseaux de communication sans fil (GSM, satellites, etc). De plus en plus d'utilisateurs mobiles veulent pouvoir utiliser leurs applications de la même manière qu'en environnement fixe et cela, quel que soit le volume de données à traiter et quel que soit le temps de calcul demandé. L'utilisation de telles applications à partir de mobiles posent néanmoins certains problèmes. D'un côté, ces applications sont coûteuses en ressources système et/ou réseau, et de l'autre côté, le mobile offre peu de ressource et celles-ci sont fluctuantes. Nous proposons une approche consistant en une distribution adaptative des applications permettant d'utiliser dynamiquement les ressources de l'environnement pour pallier l'insuffisance des ressources du mobile. De plus, face à la diversité des politiques de distribution existantes, nous proposons un cadre général de conception permettant une utilisation dynamique des politiques en fonction de l'environnement.
MOT-CLÉS : *Informatique nomade, informatique répartie, systèmes adaptatifs, approche générique, politiques dynamiques de distribution.*

ABSTRACT— Mobile computing is a domain in great expansion. Wireless networks (GSM, satellite, etc) and Portable Information Appliances (laptops, PDAs, cellular phones, etc) are developing very rapidly. More and more mobile users would like to perform their applications with the same facility as on their desktop station. Use of such applications in a mobile environment raises new challenges. Some of these applications are extremely costly in system and network resources, whereas PIAs resources are poor and wireless networks offer a very variable quality of connection. In this paper, we propose an adaptive and dynamic distribution of applications on the local environment to overcome the poorness of available resources on PIAs, and to reduce and regulate variability effects. Moreover, due to the variety of distribution policies, we propose a framework providing dynamic distribution policies.
KEY WORDS : *Mobile Computing, Distributed Computing, Adaptive Systems, Generic Approach, Dynamic Distribution Policies.*

1. INTRODUCTION

L'informatique nomade est un domaine en plein essor qui profite des percées technologiques dans le domaine des réseaux de communication sans fil (GSM, satellites, etc) et dans le domaine de l'informatique mobile (ordinateurs portables, assistants personnels, téléphones portables, etc) [Jones 99]. Dans ce contexte, l'utilisateur d'un mobile veut pouvoir utiliser son environnement de travail et ses applications multimédia de la même manière que sur sa station fixe. La plupart des applications multimédia sont interactives (éditeurs de texte, butineurs, etc) et extrêmement coûteuses en ressources système et réseau (visionneuses d'images, lecteurs audio et vidéo, visio-conférences, etc). Le cadre mobile pose quelques problèmes quant à l'utilisation de ce type d'applications :

- Due à des limitations de taille et de poids, le mobile offre peu de ressources (par rapport à une station fixe [Satyanarayanan 96]) et celles-ci sont susceptibles de varier pour satisfaire, par exemple, une politique d'économie de la batterie.
- Les réseaux sans fil offrent, en général, une bande passante plus faible que les réseaux filaires. De plus, cette bande passante est sujette à des variations importantes ainsi qu'à de fréquentes déconnexions dues aux interférences avec l'environnement, aux changements entre cellules du réseau ou à la perte de la couverture réseau.
- L'environnement du mobile change aussi suite aux déplacements. Ces changements s'opèrent au niveau matériel avec l'accès (ou la disparition de l'accès) à un certain nombre de stations (mobiles ou non) et de dispositifs (imprimantes, scanner, etc). Ces changements s'opèrent aussi au niveau logiciel avec l'apparition (ou la disparition) de fonctionnalités logicielles allant du niveau système (anticipation des accès au moyen de caches, etc) au niveau applicatif (serveur météo, trafic routier, etc).

Dans ce cadre, nous proposons une approche consistant en une distribution adaptative des applications exécutables depuis le mobile. Nous détaillons, dans le paragraphe 2, comment notre approche répond aux problèmes posés par la mobilité. Ensuite, nous présentons notre système AeDEn dans le paragraphe 3 : son architecture, l'adaptation de la distribution aux besoins de l'application, le dynamisme de la distribution et le dynamisme des politiques. Le paragraphe 4 présente notre prototype, conçu à l'aide du système MolèNE, avec un exemple de politique dynamique et les premiers résultats. Dans le paragraphe 5, nous présentons les travaux apparentés dans le domaine de l'informatique nomade et dans le domaine des systèmes distribués et enfin, nous concluons en donnant les travaux futurs envisagés dans le cadre de ce travail.

2. TERMINOLOGIE ET MOTIVATIONS

Par la suite, nous parlerons de l'*environnement* du mobile comme étant constitué de l'ensemble des *données*, *ressources* et *services* présents dans la cellule du mobile (cellule de réseau GSM, réseau WaveLAN ou réseau ad-hoc). Le terme ressources s'appliquera aux éléments matériels du mobile (processeur, mémoire, disques, imprimante, etc) ainsi qu'aux éléments matériels de l'environnement (bande passante réseau, scanner, fax, etc). Le terme service s'appliquera aux composants logiciels au niveau du système (service de nommage, d'anticipation des accès, etc) ou au niveau de l'application (service météo, commerce électronique, service de compression d'images, etc). On considérera l'*application* comme ayant été construite suivant le modèle classique d'un ensemble de *composants* communicants. Chaque composant connaît l'ensemble des données, ressources et services dont il a besoin.

Nous avons observé que la plupart des approches dans le domaine de la mobilité utilisent des ressources extérieures au mobile soit pour assurer du stockage d'informations [Kumar 95], soit pour effectuer des traitements [Noble 97, Fox 98] ou encore pour effectuer de la recherche d'informations [Gray 96, Delord 98]. Nous proposons une approche qui consiste à généraliser cette utilisation des ressources extérieures au moyen de la distribution. Notre approche consiste donc à ce que le mobile (i) connaisse les différentes ressources et services de son environnement, (ii) distribue ensuite ses applications selon leurs besoins et de manière à profiter de cet environnement, (iii) et enfin réagisse dynamiquement en adaptant la distribution et les politiques de distribution selon les changements de l'environnement.

La distribution nous permet de surmonter les inconvénients liés à la mobilité en améliorant de manière globale la qualité de service en terme de :

- **Qualité d'interactivité utilisateur :** la qualité de l'interactivité utilisateur se mesure au moyen du temps de réponse du système ou des applications. Ce temps de réponse dépend des contraintes physiques inhérentes au mobile : la mémoire, le processeur, la taille disque et la bande passante réseau. La distribution permet de mettre sur le mobile les composants de l'application en interaction avec l'utilisateur et les autres composants sont exécutés dans l'environnement. On obtient donc (i) une diminution de l'utilisation des ressources et des services du mobile, (ii) une meilleure utilisation des ressources et des services de l'environnement permettant d'optimiser les traitements, (iii) un moyen de surmonter les connexions/déconnexions puisque les traitements peuvent être effectuées hors-ligne, les résultats étant envoyés lors de la reconnexion.
- **Qualité des données :** plusieurs éléments constituent les données : le texte, les images, les sons, la vidéo, etc. Ces données sont souvent amenées à subir des transformations pour des raisons d'économie de bande passante ou d'adaptation à l'affichage du mobile [Wir 99]. En utilisant les ressources et services de l'environnement, les transformations peuvent être d'une meilleure qualité tout en satisfaisant le même temps de réponse, ou alors elles peuvent être effectuées plus rapidement tout en satisfaisant la même qualité.
- **Accessibilité :** la distribution permet d'utiliser les différentes ressources et services de l'environnement de manière transparente pour l'application par un placement approprié de ses composants. Ce placement change dynamiquement, par exemple, lorsque des ressources et services apparaissent et disparaissent dans l'environnement.

De nombreux algorithmes de distribution existent [Bernard 96] et ont été mis en œuvre dans différents systèmes [Litzkow 90, Bernard 91, Folliot 94, Cabillic 96, Namyst 96]. Ces algorithmes ont été conçus pour optimiser certains critères prédéfinis comme, par exemple, l'utilisation du processeur. Cependant, ils ne peuvent s'appliquer tels quels dans un environnement sans fil dont le caractère dynamique et la rapide évolution de la technologie rend nécessaire la prise en compte de nouveaux critères. Construire un algorithme prenant en compte tous les changements qui pourraient intervenir nous paraît irréaliste et peu extensible. C'est pourquoi, nous ne proposons pas un algorithme de distribution mais un cadre général de conception, appelé AeDEn, dans lequel les concepteurs d'applications peuvent utiliser les politiques de distribution existantes ou insérer leurs propres politiques selon les besoins de leur application. En outre, le changement dynamique de politique est possible de manière à tenir compte des changements intervenant dans l'environnement.

3. L'ENVIRONNEMENT DE DISTRIBUTION AeDEn *(Adaptive Distribution Environment)*

Traditionnellement, un algorithme de distribution se décompose en trois fonctionnalités [Zhou 88] : (i) *une politique d'information* qui définie la nature des informations collectées et la manière dont ces informations sont collectées et mises à jour, (ii) *une politique d'éligibilité* qui utilise ces informations pour élire les composants qui doivent être placés, migrés ou suspendus et (iii) *une politique de placement* qui détermine la station sur laquelle les composants élus doivent être transférés.

3.1. Architecture

AeDEn met en place ces trois fonctionnalités au sein de différents services. Ce découpage en services est effectué pour des raisons d'extensibilité et de réutilisation. En effet, un service doit pouvoir être étendu avec une nouvelle politique sans que cela n'influe sur les autres services. De plus, chacun des services peut être utilisé indépendamment, à d'autres fins que la distribution. L'application peut, par exemple, consulter la ressource bande passante et déclencher une détection des variations et cela indépendamment de la distribution.

Les trois fonctionnalités sont réparties dans les services comme le présente la figure 1. Les politiques d'élection et de placement sont implantés dans *le service de distribution*. La politique d'information est partagée entre *le service de gestion de l'environnement (politique de gestion de l'état de l'environnement)* qui se charge du mode de collecte et du stockage des informations concernant l'état des ressources et services de l'environnement, et *le service de détection et notification (politique de détection et notification de variations)* qui se charge de détecter et de surveiller les variations de l'ensemble des ressources et services.

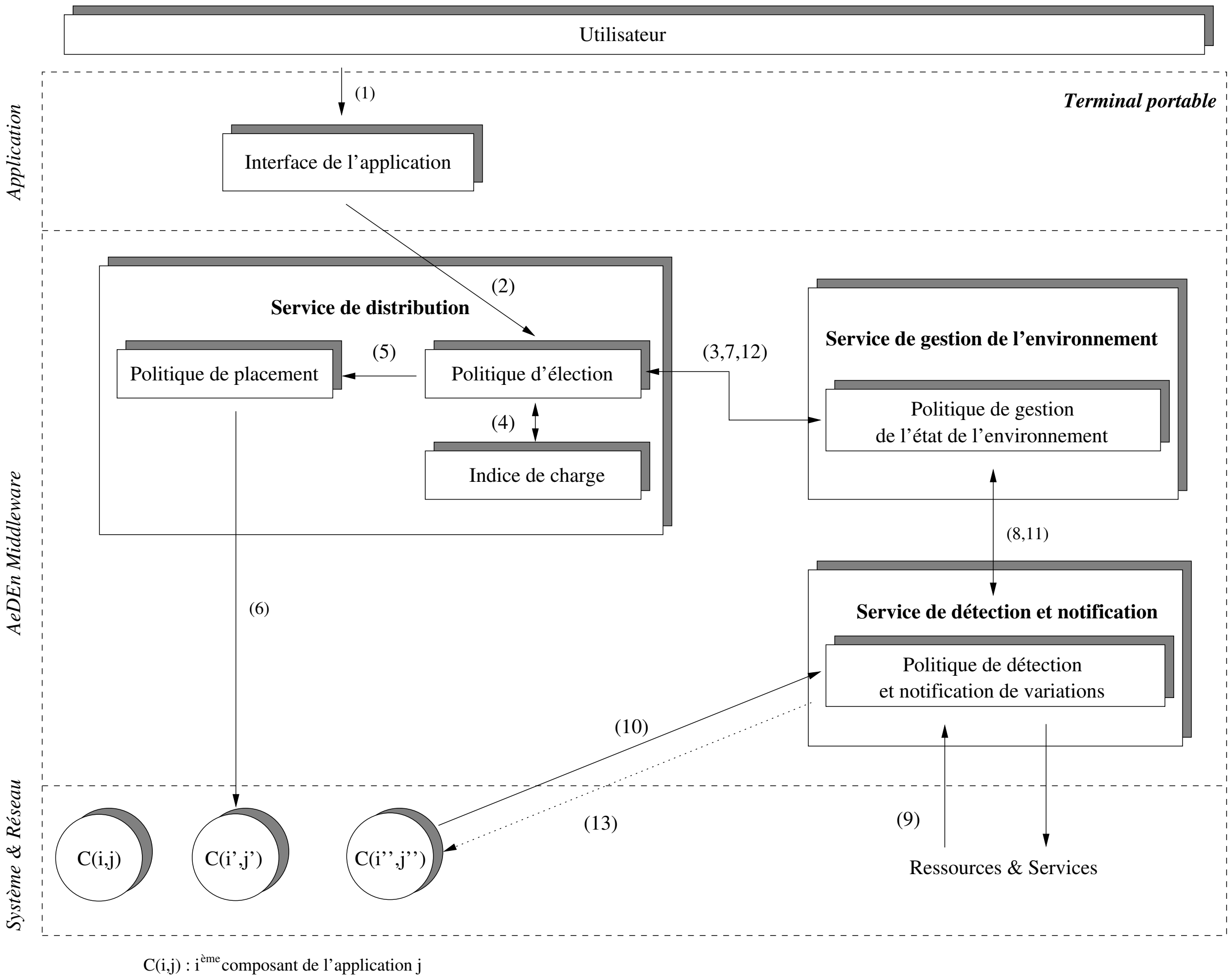

Figure 1. *Architecture d'AeDEn*

3.2. Distribution adaptée à l'application

Au lancement d'une application, les interactions entre cette application et AeDEn constituent la première phase d'adaptation. Celle-ci consiste à prendre en compte les besoins des composants de l'application ainsi que les ressources et services disponibles pour effectuer la distribution la plus adéquate des composants.

La figure 1 illustre les différentes étapes de ce processus. L'application fournit au service de distribution, et plus particulièrement à la politique d'élection, l'ensemble des composants pouvant être distribués (1,2). La politique d'élection interagit avec le service de gestion de l'environnement pour connaître l'état des ressources et services (3). L'indice de charge peut être calculé (4). Les composants élus sont alors soumis à la politique de placement (5) qui détermine la station appropriée et effectue le transfert sur celle-ci (6). Le service de distribution enregistre enfin les composants dans le service de gestion de l'environnement (7) et active une détection de variation sur ces composants (8).

Les composants de l'application sont ainsi placés et le système peut tenir compte des changements de l'environnement pour reconsidérer ses choix.

3.3. Distribution dynamique en fonction de l'environnement

Les changements dans l'environnement sont pris en compte par les services d'AeDEn à deux niveaux différents: au niveau de l'application et au niveau du système de distribution en lui-même. Dans le premier cas, en fonction des changements de l'environnement et des décisions des différentes politiques, le placement des composants de l'application est dynamiquement recalculé et réeffectué. Le placement dynamique est décrit dans le paragraphe 3.3.1. Dans le deuxième cas, les changements de l'environnement influent sur les politiques de distribution elles-mêmes et conduisent à un changement dynamique des politiques utilisées. Nous détaillons le dynamisme des politiques dans le paragraphe 3.3.2.

3.3.1. Placement dynamique

Les différentes interactions intervenant au cours du placement dynamique sont illustrées sur la figure 1. Lors d'une variation soit sur les ressources ou services (9) soit sur les composants (une disparition par exemple) (10), le service de détection et notification notifie le service de gestion de l'environnement qui met à jour sa base d'informations (11) et notifie le service de distribution si des composants distribués sont concernés[1] (12). La politique d'élection détermine alors si ces changements affectent le placement et effectue, au besoin, un nouveau placement (3,4,5,6) qui peut impliquer de nouveaux composants, la suspension et/ou la migration des composants existants.

3.3.2. Politiques dynamiques

Comme nous l'avons décrit dans le paragraphe 2, une politique particulière peut être plus ou moins pertinente suivant les besoins et contraintes de l'application et suivant la disponibilité des ressources et services de l'environnement. Pour qu'AeDEn puisse répondre aux différents besoins en matière de distribution, nous introduisons le concept de *politique dynamique*. Une politique dynamique peut changer dynamiquement de variante, par exemple lorsque des changement surviennent dans l'environnement (voir figure 2). Chaque politique dynamique possède une seule variante à un instant t mais peut dynamiquement en changer et en avoir une différente à un instant $t+1$.

Ce mécanisme de politique dynamique est utilisé par chacune des politiques des services d'AeDEn. Les différentes variantes des politiques de distribution existantes peuvent donc être implantées et mise à disposition des applications. Un éventail des différentes variantes des politiques existantes sera présenté dans le paragraphe 5. Les applications peuvent également implanter leur propre variante de politique et l'inclure dynamiquement. Nous utilisons également ce mécanisme pour l'indice de charge. Il peut alors changer dynamiquement et prendre en compte les critères de la mobilité comme la faible bande passante,

[1]Les composants de l'application peuvent aussi avoir été conçus pour s'adapter en fonction des changements de l'environnement. Dans ce cas, ils en sont directement notifiés par le service de détection et notification (13).

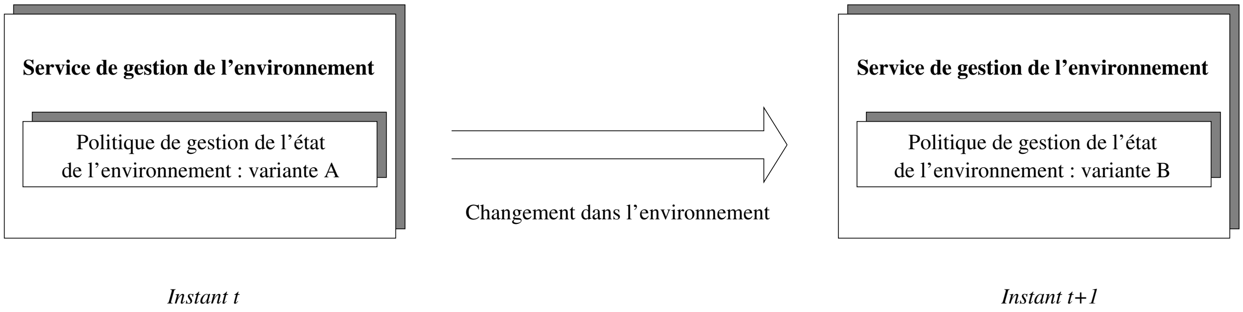

Figure 2. *Exemple de politique dynamique dans le cas du service de gestion de l'environnement*

la probabilité de déconnexion et le temps de vie de la batterie.

4. Prototype d'AeDEn

Nous avons utilisé un cadre général de gestion des environnements mobiles MolèNE pour implanter un premier prototype d'AeDEn. Chaque service d'AeDEn est implanté par un ensemble de composants MolèNE. Le paragraphe suivant présente l'architecture d'un composant MolèNE. Nous illustrons, dans le paragraphe 4.2, l'utilisation des composants MolèNE pour implanter la politique d'élection. Enfin, nous présentons, dans le paragraphe 4.3, quelques résultats préliminaires sur la faisabilité de notre approche.

4.1. Notion de composant MolèNE

Grâce aux techniques de réflexion et d'introspection des langages orientés-objet, nous avons construit MolèNE [Segarra 00], un cadre général qui permet de changer dynamiquement l'implantation d'un composant (la plus petite unité logicielle dans MolèNE). Ce mécanisme repose sur une stratégie d'adaptation qui décide des changements d'implantation et sur un objet *Controller* qui les exécute. La figure 3 montre l'ensemble des classes constituant un composant MolèNE.

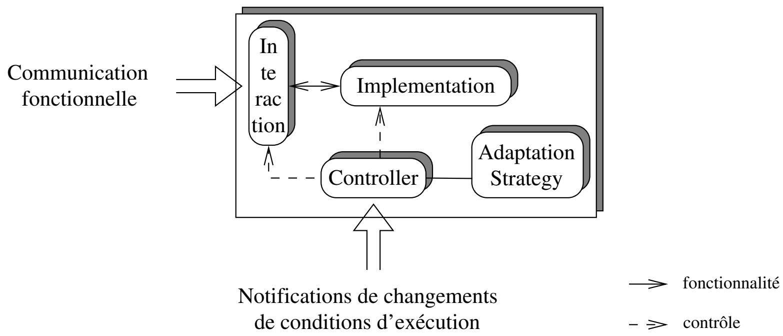

Figure 3. *Structure d'un composant MolèNE*

L'objet *Interaction* reçoit des demandes de service et les transmet à l'objet *Implementation* courant qui assure la fonctionnalité du composant. Le *Controller* reçoit des notifications de changements dans l'environnement et décide du remplacement de l'objet *Implementation* courant selon les informations contenues dans l'objet *Adaptation Strategy*. Lors de ce remplacement, l'objet *Implementation* construit une représentation des informations qu'il utilise et que nous appelons état interne. Cet état interne est utilisé pour initialiser le nouvel objet *Implementation*.

4.2. Exemple: politique d'élection

La figure 4 illustre la mise en œuvre de la politique d'élection en utilisant des composants MolèNE.

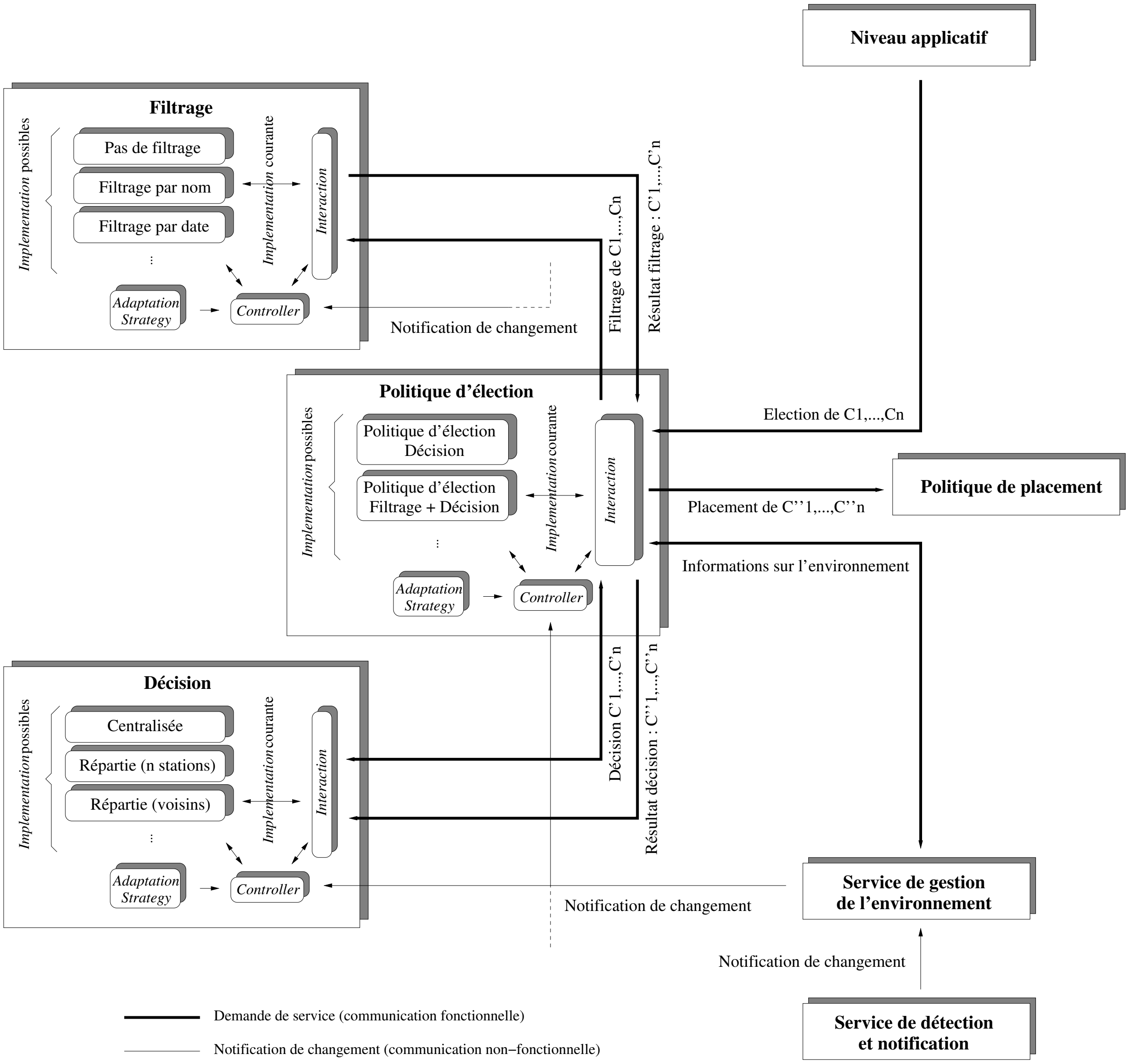

Figure 4. *Implantation de la politique d'élection à l'aide de composants MolèNE*

L'application soumet ses composants à la politique d'élection. L'objet *Implementation* courant de la politique d'élection (*Politique d'élection avec Filtrage et Décision* sur la figure 4) invoque alors les composants MolèNE dont il a besoin : un composant MolèNE de filtrage permettant d'éliminer du processus de distribution les objets de trop courte durée de vie, et un composant MolèNE de prise de décision. Ces deux composants peuvent avoir plusieurs objets *Implementation* possibles : le filtrage peut se faire suivant le nom de l'objet, sa date de lancement, etc et la prise de décision peut se faire de manière centralisée, avec n stations, avec ses voisins, etc. La politique d'élection a aussi besoin de services existants comme le service de gestion de l'environnement ou la politique de placement.

Les notifications de changements dans l'environnement sont envoyées par le service de gestion de l'environnement (après notification par le service de détection et notification) au *Controller* de la politique d'élection et aux *Controller* des fonctionnalités dépendantes de la politique d'élection. Ces notifications indiquent que l'objet *Implementation* courant n'est peut-être plus adéquat. L'objet *Implementation* courant est alors changé par le *Controller* suivant l'objet *Adaptation Strategy* fourni par l'application.

Pour illustrer l'intérêt de ce changement dynamique de politiques d'élection, prenons l'exemple d'un utilisateur avec un ordinateur portable. Tant que cet ordinateur portable reste connecté à un réseau local avec une liaison Ethernet, le débit réseau permet d'avoir une politique de décision répartie impliquant n stations du réseau local. Ensuite, si cet ordinateur se déplace tout en gardant une connexion de type sans fil, il est préférable d'adopter une politique de décision centralisée permettant de réduire le trafic réseau et coïncidant mieux avec la structure en point d'accès. Enfin, si cet ordinateur se connecte à un réseau ad-hoc, une politique de décision répartie mais n'impliquant que les ordinateurs voisins permettrait une meilleure conformité avec la topologie réseau tout en ne surchargeant que localement les communications.

4.3. Premiers résultats

Les différentes politiques de distribution ont déjà été largement étudiées en termes de mesures de performance, de coût de migration, etc. Notre approche introduit un nouveau facteur qui est la possibilité de remplacer une politique par une autre. Nous avons examiné le coût de cette opération en terme de bande passante et de temps de réponse. Pour ce faire, nous avons évalué le prototype d'AeDEn que nous avons implanté en Java. Toutes les mesures ont été réalisées avec, comme environnement mobile, un IBM ThinkPad pourvu d'une connexion WaveLAN 1,6Mb/s et utilisant, comme environnement fixe, une station Sun Ultra-1 et une station Sun Ultra-60 avec une connexion Ethernet 100Mb/s.

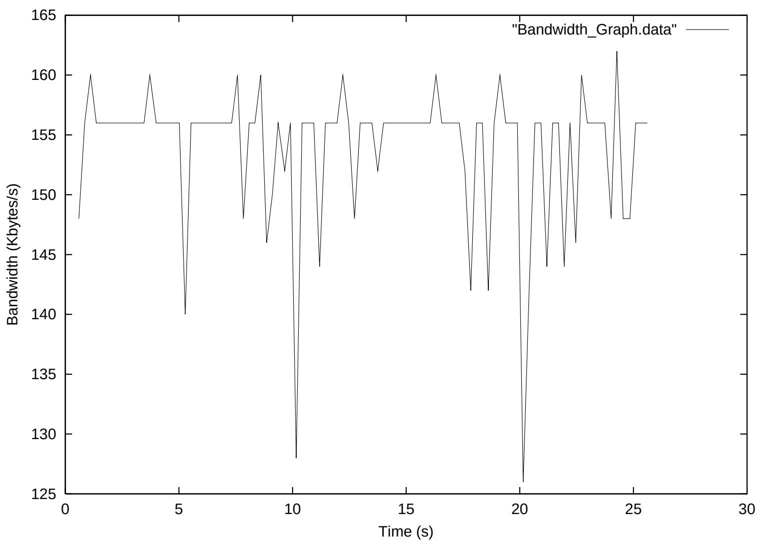

Figure 5. *Variations de bande passante lors du passage d'une politique d'information centralisée à une politique d'information répartie*

Nous avons testé les variations de bande passante produite par le passage d'une politique d'information centralisée à une politique d'information répartie. Les informations échangées par la politique d'information étaient constituées de l'état de la bande passante, l'état de charge du processeur et l'état de la batterie pour le mobile. Au départ, la station Ultra-60 fait office de serveur d'information et les autres jouent le rôle de clients et ensuite, elles passent toutes en mode réparti. La figure 5 montre les variations de bande passante : à l'instant $t = 10$, les stations passent en mode réparti et reçoivent les informations de l'environnement et à l'instant $t = 20$, elles repassent de nouveau en mode client/serveur et envoient les informations de l'environnement au serveur. On peut observer que la bande passante utilisée représente moins de 20% de la bande passante totale. De plus, cette variation est comparable aux variations induites par la connexion sans fil.

Nous avons aussi mesuré le temps de réponse pour passer d'une implantation à une autre. Le mécanisme de changement prend en lui-même $5ms$ ($25ms$ lors du premier appel pour cause de chargement de la classe). Dans le cas de la politique d'information, la transmission de l'état de l'environnement prend en tout entre 275 et $295ms$ selon les variations de la bande passante.

5. TRAVAUX APPARENTÉS

Peu d'approches proposent l'utilisation de ressources distantes pour pallier la pauvreté des ressources des mobiles. Quelques approches utilisent le modèle agent [Gray 96]. Une agent mobile est un programme actif capable de se déplacer de manière autonome à travers le réseau et d'interagir avec les ressources et services de l'endroit où il se trouve. Ce modèle permet de déléguer des actions comme la collecte et le filtrage d'informations aux agents se trouvant sur le réseau fixe et de ne recevoir que les résultats sur le portable. Par exemple, Alycta [Delord 98] fournit un accès au Web à travers le GSM en déléguant le chargement des documents et leur dégradation à un agent effectuant ces tâches hors-ligne.

Dans le modèle *Client/Proxy/Server*, un intermédiaire est exécuté sur une station fixe et utilise les ressources de cette station pour fournir des fonctionnalités à la station mobile. Ces fonctionnalités peuvent être des protocoles réseau adaptés [Liljeberg 96], le stockage de résultats lors de déconnexions, le filtrage de données comme la compression, la dégradation ou la conversion vers un média adapté [Wir 99].

Plusieurs intermédiaires peuvent aussi être utilisés de manière distribué pour accroître le nombre de ressources disponibles. Par exemple, des techniques de distribution sont utilisées dans Daedalus [Fox 98] pour déployer des services d'adaptation (*TACC servers*) sur un cluster de PC. La politique employée consiste à augmenter le nombre d'intermédiaires suivant le nombre de requêtes en attente. Cette approche est néanmoins spécifique : les processus distribués sont d'un type unique, défini par le système d'adaptation pour une tâche particulière. De plus, la politique de distribution est figée et ne peut pas prendre en compte de nouveaux critères. Pour pouvoir distribuer une application d'une manière générale, les techniques de distribution existantes dans le domaine des systèmes distribués doivent être considérées.

Il existe de nombreux algorithmes de distribution qui assurent de bonnes performances, l'équilibrage de charge et la stabilité. Une classification des principaux algorithmes [Bernard 96] est présentée dans la figure 6. Ces différents algorithmes ont été expérimentés dans différents systèmes comme Radio [Bernard 91] et Utopia [Zhou 91] pour les algorithmes non-adaptatifs et non-préemptifs, Condor [Litzkow 90] pour les algorithmes préemptifs, GatoStar [Folliot 94] pour le placement et la migration, PM^2 [Namyst 96] et Startdust [Cabillic 96] pour les algorithmes adaptatifs.

Ces systèmes effectuent un placement reposant uniquement sur le critère de charge du processeur. Il existe également des approches multi-critères [Zhou 91, Folliot 92] mais elles ne prennent pas en compte les nouveaux critères introduits par la mobilité comme une bande passante faible et hautement variable, les probabilités de déconnexion et de reconnexion ou encore le temps de vie de la batterie.

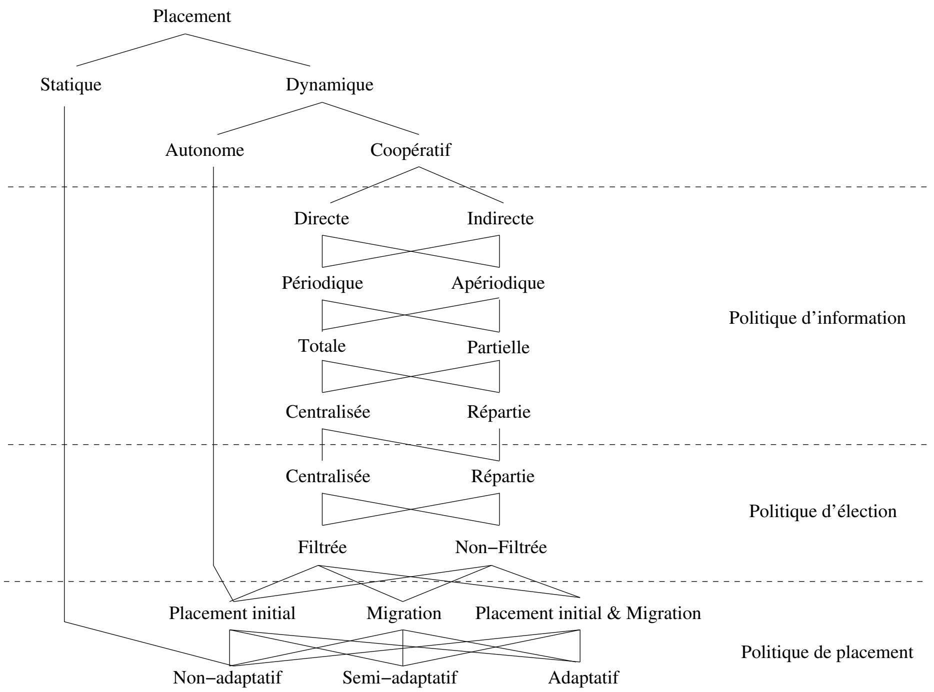

Figure 6. *Classification des algorithmes de placement*

6. CONCLUSION

Dans cet article, nous avons présenté AeDEn, un système fournissant des mécanismes adaptatifs et dynamiques de distribution des applications dans des environnements hautement variables. Cette distribution permet de pallier l'insuffisance des ressources disponibles sur les terminaux portables en utilisant les ressources et services de l'environnement. Nos mécanismes sont adaptatifs dans les sens où ils prennent en compte les besoins et contraintes des applications. Ils sont dynamiques dans les sens où ils prennent en compte les changements de l'environnement. De plus, ces changements sont pris en compte à deux niveaux : (i) par le placement dynamique des composants de l'application et (ii) par la modification des politiques de distribution elles-mêmes.

Actuellement, nous travaillons à l'identification des critères de placement pertinents et stratégies de changement de politiques appropriées en environnements mobiles. Nous travaillons aussi sur la cohérence des différentes politiques. En effet, changer une politique sur une station affecte le comportement des autres stations et ne doit pas conduire à des états incohérents. Enfin, nous envisageons de tester notre système avec une application de presse électronique. Cette application nous semble pertinente pour tester nos techniques de distribution puisqu'elle manipule un volume important de données et requiert de la puissance de calcul.

7. BIBLIOGRAPHIE

[Bernard 91] Guy Bernard, Dominique Stève, et Michel Simatic. Placement et migration de processus dans les systèmes répartis faiblement couplés. *Technique et Science Informatiques*, 10(5), May 1991.

[Bernard 96] Guy Bernard et Bertil Folliot. Caractéristiques Générales du Placement Dynamique : Synthèse et Problématique. Dans *Tutoriel invité dans les actes de l'école d'été MASI-IMAG-INT-*

PRISM "Placement dynamique et répartition de charge : application aux systèmes parallèles et répartis", Presqu'île de Giens, France, July 1996.

[Cabillic 96] Gilbert Cabillic et Isabelle Puaut. Dealing with Heterogeneity in Stardust: An Environment for Parallel Programming on Networks of Heterogeneous Workstations. Dans *Proceedings of 2nd International Euro-Par Conference (Euro-Par'96)*, pp. 114–119, Lyon, France, August 1996.

[Delord 98] Xavier Delord, Stéphane Perret, et Andrzej Duda. Efficient Mobile Access to the WWW over GSM. Dans *Proceedings of the 8th ACM SIGOPS European Workshop Support for Composing Distributed Applications*, pp. 1–6, Sintra, Portugal, September 1998.

[Folliot 92] Bertil Folliot. *Méthodes et Outils de Partage de Charge pour la Conception et la Mise en Œuvre d'Applications dans les Systèmes Réparties Hétérogènes*. PhD thesis, MASI laboratory, Paris 6 University, France, 1992.

[Folliot 94] Bertil Folliot et Pierre Sens. GATOSTAR: A Fault Tolerant Load Sharing Facility for Parallel Applications. Dans K. Echtle, D. Hammer, et D. Powell, éditeurs, *Proceedings of the 1st European Dependable Computing Conference*, vol. 852 *Lecture Notes in Computer Science*. Springer-Verlag, October 1994.

[Fox 98] Armando Fox, Steven D. Gribble, Yatin Chawathe, et Eric A. Brewer. Adapting to Network and Client Variation Using Infrastructural Proxies: Lessons and Perspectives. *IEEE Personal Communications (invited submission)*, 5(4):10–19, August 1998.

[Gray 96] Robert Gray, David Kotz, Saurab Nog, Daniela Rus, et George Cybenko. Mobile agents for mobile computing. Rapport Technique PCS-TR96-285, Department of Computer Science, Dartmouth College, May 1996.

[Jones 99] Alan Jones. Mobile Computing to Go. *IEEE Concurrency*, 7(2):20–23, April 1999.

[Kumar 95] Puneet Kumar et M. Satyanarayanan. Flexible and Safe Resolution of File Conflicts. Dans *Proceedings of the USENIX Winter 1995 Technical Conference*, New Orleans, Louisiana, USA, January 1995.

[Liljeberg 96] Mika Liljeberg, Heikki Helin, Markku Kojo, et Kimmo Raatikainen. Enhanced Services for World-Wide Web in Mobile WAN Environment. Dans *Proceedings of the IEEE Global Internet 1996 Conference (revised version)*, London, England, November 1996.

[Litzkow 90] Michael J. Litzkow et Miron Livny. Experience With The Condor Distributed Batch System. Dans *Proceedings of the IEEE Workshop on Experimental Distributed Systems*, pp. 97–101, Huntsville, Alabama, USA, October 1990.

[Namyst 96] R. Namyst et J.-F. Méhaut. PM^2: Parallel Multithreaded Machine. A Computing Environment for Distributed Architectures. Dans E. H. D'Hollander, G. R. Joubert, F. J. Peters, et D. Trystram, éditeurs, *Parallel Computing: State-of-the-Art and Perspectives, Proceedings of the Conference ParCo'95, 19-22 September 1995, Ghent, Belgium*, vol. 11 *Advances in Parallel Computing*, pp. 279–285, Amsterdam, February 1996. Elsevier, North-Holland.

[Noble 97] Brian D. Noble, M. Satyanarayanan, J.E. Tilton, J. Flinn, et K.R. Walker. Agile Application-Aware Adaptation for Mobility. Dans *Proceedings of the 16th ACM Symposium on Operating Systems Principles (SOSP'16)*, Saint-Malo, France, October 1997.

[Satyanarayanan 96] M. Satyanarayanan. Fundamental Challenges in Mobile Computing. Dans *Proceedings of the 5th Annual ACM Symposium on Principles of Distributed Computing (PODC'96)*, Philadelphia, Pennsylvania, USA, May 1996.

[Segarra 00] M.T. Segarra et F. André. A Framework for Dynamic Adaptation in Wireless Environments. Dans *Proceedings of the Technology of Object-Oriented Languages and Systems (TOOLS'2000)*, Saint Malo, France, June 2000.

[Wir 99] Wireless Internet Today. *Wireless Application Protocol - White Paper*, October 1999.

[Zhou 88] S. Zhou. A Trace-driven Simulation Study of Dynamic Load Balancing. *IEEE Transactions on Software Engineering*, 14(9):1327–1341, September 1988.

[Zhou 91] S. Zhou, J. Wang, X. Zheng, et P. Delisle. Utopia: A Load Sharing System for Large, Heterogeneous Distributed Computer Systems. Rapport Technique 257, Computer Systems Research Institute, Toronto University, December 1991.

∞

Biographies

Frédéric Le Mouël *est titulaire d'un diplôme en informatique et télécommunication (DIIC 1997), et est actuellement doctorant au sein du projet SOLIDOR à l'IRISA. Ses domaines de recherche concernent l'informatique nomade et l'informatique répartie. Il étudie en particulier l'utilisation de techniques de distribution prenant en compte les contraintes de l'informatique nomade.*

Françoise André *est professeur à l'IFSIC, Université de Rennes 1 et exerce son activité de recherche à l'IRISA dans le projet SOLIDOR. Jusqu'en janvier 1996 son thème de recherche a concerné la conception et le développement d'environnements de programmation et d'exécution pour calculateurs parallèles et distribués, notamment dans le cadre du projet PAMPA dont elle a été la responsable. Les résultats les plus significatifs sont le développement de l'environnement de compilation PANDORE et la participation au projet européen PREPARE. Depuis 1996 ses activités de recherche prennent en considération deux évolutions majeures en informatique : les réseaux sans fil d'une part, l'Internet de l'autre part. Son objectif est de concevoir une couche middleware placée entre un système d'exploitation classique et les applications pour traiter des problèmes spécifiques à un environnement mobile (système MolèNE). Plusieurs types d'applications sont examinés, en particulier les systèmes transactionnels sur l'Internet (application METIS, projet européen Agentisme, projet Charcot).*