

# Vers une approche d'analyse et de diagnostic pour les architectures M2M autonomes et sensibles au contexte

Imene Lahyani<sup>\*,\*\*</sup>, Lamia Ben Amor<sup>\*\*\*</sup>, Ismael Bouassida Rodriguez<sup>\*,\*\*</sup>,  
Mohamed Jmaiel<sup>\*\*\*</sup>

<sup>\*</sup>CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France

<sup>\*\*</sup>Univ de Toulouse, LAAS, F-31400 Toulouse, France

<sup>\*\*\*</sup>University of Sfax, ReDCAD, Sfax, Tunisia

iabdenna@laas.fr, lamia.benamor@redcad.org

bouassida@laas.fr, mohamed.jmaiel@rnu.enis.tn

**Résumé.** L'évolution incontournable des systèmes ubiquitaires et en particulier les systèmes machine-to-machine (M2M) provoque la complexité de la gestion de ce type de système. La conception d'un tel système autonome et sensible au contexte basé sur une architecture d'auto-adaptabilité représente une solution prometteuse. Cette architecture modifie le comportement du système en fonction des besoins des utilisateurs, des ressources disponibles et de l'environnement de l'application. Dans cet article, nous proposons une approche d'analyse et de diagnostic des systèmes M2M en se basant sur des méthodes statistiques qui visent, en premier lieu, à détecter les changements de contexte (phase d'analyse). En deuxième lieu, le module de diagnostic permet d'identifier les causes des changements de contexte détectés durant la phase d'analyse et fournir à la phase de planification les informations nécessaires afin d'exécuter les actions d'adaptation appropriées.

## 1 Introduction

Au cours des dernières années, l'informatique ubiquitaire, représentant l'un des enjeux de demain, a évolué à une vitesse exceptionnelle permettant ainsi de connecter un nombre important d'objets hétérogènes (capteurs, actionneurs, Smartphone, application, etc.). En effet, l'objectif de l'informatique ubiquitaire est d'intégrer les technologies informatiques dans la vie quotidienne de l'homme le plus simplement possible. Un des principaux concepts de l'informatique ubiquitaire est le Machine-to-Machine (M2M).

Le M2M selon (Nabet et al. (2006)) est "l'association des technologies de l'information et de la communication, avec des machines communicantes dans le but de fournir à ces dernières les moyens d'interagir sans intervention humaine". Ce concept fleurisse dans de nombreux domaines : gestion de l'énergie, domotique, gestion de flottes, eSanté, télésurveillance, etc.

Ainsi, un système M2M désigne littéralement tous les systèmes intégrant un grand nombre de machines communicantes de nature hétérogènes ayant des capacités limitées, des différents réseaux de communication, des serveurs et une multitude d'applications mobiles ayant des environnements hautement dynamiques.

Cette cohabitation assez hétérogène de machines, de réseaux et d'applications engendre l'apparition des changements de contexte dans ce type de système.

Alors, gérer les changements de contexte de manière autonome et de réagir en conséquence représente un défi majeur. Il est donc nécessaire de concevoir une architecture M2M autonome et sensible au contexte qui modifie son comportement en fonction des besoins des utilisateurs, des ressources disponibles et de l'environnement de l'application.

En outre, un système sensible au contexte est caractérisé par une boucle comportant quatre phases à savoir : monitoring, analyse, planification et exécution. Cette boucle met en place une architecture d'auto-adaptation.

Dans ce contexte, plusieurs travaux de recherche ont été proposés comme (Zheng et al., 2006), (Lupu et al., 2008), (Bouassida Rodriguez et al., 2009), (Cioara et al., 2010), (Alaya et Monteil, 2012). Ces approches offrent aux systèmes ubiquitaires la possibilité de détecter les changements de contexte qui ont eu lieu et de réagir vis-à-vis des détections afin d'ajuster le comportement de ces systèmes en conséquence. Malgré les avantages offerts par ces approches, ces dernières peuvent mener le système à des situations néfastes suite, d'une part, à un mauvais analyse (l'utilisation des seuils fixes) et d'autre part suite à un manque de diagnostic. Ainsi, le passage par une étape de diagnostic ayant pour but l'identification des causes des changements de contexte détectés s'avère nécessaire. Le diagnostic, alors, fournit à la planification les informations nécessaires pour générer les actions de reconfigurations adéquates afin de mieux ajuster le comportement du système face aux changements de contexte. De plus, selon la littérature, le diagnostic utilise des techniques qui peuvent être classifiées en deux catégories : les méthodes de diagnostic à base de modèle (Cordier et al. (2000), Provan (2002)) et les méthodes de diagnostic guidées par les données ou sans modèle (Zwingelstein (1995), ONDEL. (2006)).

Dans cet article, nous mettons l'accent sur une classification des paramètres de contexte proposée dans Chaari et al. (2005) afin de couvrir toutes les situations contextuelles. Ensuite, nous rappelons la phase d'analyse proposée dans des travaux antérieurs Lahyani et al. (2012). Puis, nous introduisons une approche de diagnostic sans modèle pour identifier les causes des changements de contexte détectés par la phase d'analyse en se basant sur des méthodes statistiques et générer une requête de changement à la phase de planification.

Le reste de l'article est organisé comme suit : dans la Section 2, nous présentons, en premier lieu, quelques notions de bases relatives aux systèmes M2M et à la sensibilité au contexte. En deuxième lieu, nous donnons un aperçu sur les approches d'auto-adaptabilité existantes dans les systèmes ubiquitaires et les méthodes de diagnostic existantes. La Section 3 expose l'approche d'analyse et de diagnostic proposée permettant la détection des changements de contexte dans les systèmes M2M et l'identification des causes de ces changements en se basant sur des méthodes statistiques. La Section 4 est consacrée à décrire l'étude de cas intitulée "télé-surveillance d'un restaurant intelligent". Cette étude de cas illustre la faisabilité de l'approche proposée. Enfin, ce papier est clôturé par une conclusion et quelques perspectives (Section 5).

## 2 État de l'art

Cette section débute par une présentation de quelques notions de base (système M2M, *autonomic computing*, contexte et sensibilité au contexte). Ensuite, nous donnons un aperçu sur

les approches d'auto-adaptabilité existantes dans les systèmes ubiquitaires. Enfin, nous citons les méthodes de diagnostic existantes.

## 2.1 Notions de base

### 2.1.1 Les systèmes M2M

Un système M2M consiste à l'utilisation d'une machine (capteur, actionneur, etc) afin de capturer un événement. Ce dernier sera relayé à travers un réseau de communication (fixe, mobile ou hybride) à une application. Ainsi, un système M2M représente le résultat d'une interaction entre les machines, les réseaux de communication et les applications.

En outre, ETSI (European Telecommunications Standards Institute) a élaboré un standard décrivant l'architecture des systèmes M2M. La Figure 1 présente l'architecture simplifiée de haut niveau des systèmes M2M.

Comme indiqué à la Figure 1, le système M2M permet un échange bidirectionnel d'infor-

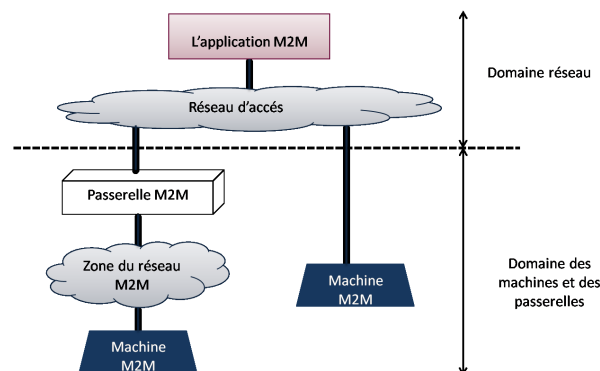


FIG. 1 – L'architecture simplifiée des systèmes M2M élaborée par ETSI

mations entre les machines M2M et l'application. De plus, cette architecture comporte deux domaines : le domaine des machines et des passerelles et le domaine réseau.

En effet, une machine M2M représente un périphérique qui exécute les applications M2M à l'aide des capacités de service qu'ils possèdent. En outre, cette machine M2M peut être connectée au domaine réseau directement via le réseau d'accès. Dans d'autres cas, les machines ne peuvent pas interagir directement avec l'application vu qu'elles ne possèdent que des capacités limitées. Dans ces cas, la relation est médiatisée par un autre périphérique (une passerelle M2M) qui permet une certaine forme de consolidation de la communication.

Cependant, la complexité des systèmes M2M a rendu difficile l'autogestion d'un tel système en fonction des changements qui peuvent intervenir dans le contexte. Pour cela, la conception d'un système M2M autonome et sensible au contexte s'avère nécessaire.

### 2.1.2 L'autogestion des systèmes informatiques

La complexité des systèmes informatiques a rendu de plus en plus difficile de gérer ces systèmes à la main. Par conséquent, IBM a proposé, en 2001, un nouveau défi appelé l'autogestion des systèmes informatiques ou *Autonomic Computing* en référence au fonctionnement du système nerveux humain.

L'autogestion des systèmes informatiques représente, ainsi, un ensemble de concepts, de technologies et d'outils permettant aux systèmes de fonctionner de manière plus autonome.

En outre, les caractéristiques d'un système autonome sont :

- Auto-configuration : Ajuster dynamiquement ses ressources à l'exécution afin de s'adapter aux changements de son état interne et de son environnement externe.
- Auto-adaptation : Détecter des problèmes et se reconfigurer de façon à assurer un fonctionnement continu.
- Auto-optimisation : Détecter la dégradation de performance du système afin de l'améliorer.
- Auto-protection : Se défendre contre les attaques intérieures/extérieures.
- Sensibilité au contexte : Avoir conscience de son environnement et être capable de réagir face aux changements.
- ouverture : Indépendant de l'hétérogénéité des systèmes.
- Anticipation : Prévoir ses besoins et ses comportements ainsi que ceux de son environnement et être capable de s'autogérer de façon proactive.

### 2.1.3 Notion de contexte et de sensibilité au contexte

Abowd et al. (1999) ont défini le contexte en précisant la nature des entités relatives au contexte comme étant "Toute information qui peut être utilisée pour caractériser la situation d'une entité. Une entité est une personne, un lieu ou un objet qui est considéré comme pertinent pour l'interaction entre un utilisateur et les applications elles-mêmes".

Cette définition est d'ordre générique. En effet, Dey explique cette généralité du fait que les paramètres de contexte peuvent être fournis directement par l'utilisateur ou par des terminaux situés dans l'environnement de l'utilisateur et de l'application.

De plus, la prise en compte du contexte d'utilisation dans les applications est un domaine de recherche d'actualité connu sous le nom de "sensibilité au contexte". Selon Dey et al, une application sensible au contexte doit percevoir la situation de l'utilisateur dans son environnement et adapter par conséquent son comportement à la situation en question.

## 2.2 Les approches d'auto-adaptabilité existantes dans les systèmes ubiquitaires

Pour assurer l'autogestion des systèmes, IBM propose une boucle de contrôle, appelée la boucle MAPE-K, décrite dans la Figure 2 ( Computing et al. (2006)).

Cette boucle comporte quatre parties partageant les mêmes *knowledge*.

- Le Monitoring vise à observer, extraire et collecter des informations sur le système géré.
- L'Analyse consiste à exploiter les informations tirées à partir de la phase de monitoring afin d'évaluer l'état du système et prévoir les situations futures.

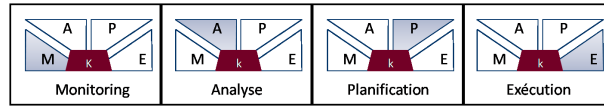


FIG. 2 – la boucle MAPE-K proposée par IBM

- La Planification prévoit les mécanismes qui construisent les actions de reconfigurations nécessaires afin d’ajuster le comportement du système.
- L’Exécution permet de contrôler l’exécution des actions de reconfigurations planifiées.
- *knowledge* permet de représenter le contexte, le système,...

Dans la littérature, plusieurs techniques ont été proposées dans le but de gérer les changements de contexte intervenant dans les systèmes ubiquitaires en se basant sur la boucle MAPE-K.

En effet, nous énumérons les travaux de (Zheng et al., 2006), (Lupu et al., 2008), (Bouassida Rodriguez et al., 2009), (Cioara et al., 2010) et (Alaya et Monteil, 2012).

Dans l’approche de (Cioara et al., 2010), la phase de Monitoring collecte des données appartenant à l’environnement du système tels que la température, l’humidité, la luminosité,...

La phase d’analyse permet la détection des modifications de contexte au niveau environnement en se basant sur des valeurs seuils fixes. Bien que cette approche permette l’autogestion des changements contextuels, elle peut mener à des fausses détections de modification à travers l’utilisation des seuils fixes. De plus, nous prenons l’exemple suivant : la phase d’analyse détecte une élévation significative de la valeur de la température d’un établissement. En faite, cette élévation peut être causée par plusieurs facteurs tels que “Fonctionnement du système de climatisation avec ouverture des fenêtres”, “la présence de plusieurs personnes dans l’établissement”, “l’éclairage”. Dans ce cas, la sélection de l’action d’adaptation pendant la phase de planification engendre des pertes de temps et nuit à la performance du système. L’idée, donc, est de diagnostiquer l’élévation de la température afin d’identifier sa cause réelle et par conséquent améliorer les performances du système.

En outre, (Bouassida Rodriguez et al., 2009) proposent un Framework, nommé *FACUS*, destiné aux systèmes collaboratifs ubiquitaires. Ce Framework fournit un service de déploiement sensible au contexte. Il permet la détection des changements qui peuvent intervenir dans le contexte en utilisant des seuils prédéfinis et le déclenchement des adaptations nécessaires en utilisant des ontologies. Cependant, (Bouassida Rodriguez et al., 2009) tiennent compte d’un seul paramètre de contexte qui est le niveau de batterie d’une machine. Pour cela, il est inutile de consacrer toute une boucle pour la gestion de ce paramètre de contexte.

Lupu et al. (2008) propose un Framework, intitulé *AMUSE*, destiné aux environnements ubiquitaires menés d’une application de eSanté afin de surveiller la santé des patients d’un hôpital ou d’une maison. Ce Framework consiste à autogérer le système de surveillance aux changements de contexte qui ont eu lieu suite à des activités des utilisateurs ou des pannes des matériels. Néanmoins, dans les applications de eSanté, une élévation de la température peut influencer sur l’état des patients et peut causer ainsi des situations néfastes. Donc, l’étape de diagnostic peut résoudre ce problème.

De plus, Alaya et Monteil (2012) suggèrent une nouveau Framework, appelé *FRAMESELF*, destiné aux systèmes ubiquitaires sensibles au contexte basé sur des ontologies et des graphes.

En conclusion, malgré les bénéfices offerts par ces approches, ces dernières peuvent mener le système à des situations néfastes causant l'échec total du système suite à un mauvais analyse ou à un manque de diagnostic. Alors, un module intermédiaire de décision délimitée par la phase d'analyse et la phase de planification s'avère nécessaire afin de bien diagnostiquer les changements de contexte intervenants dans le système. Cette phase, nommée diagnostic, consiste à identifier les causes des changements de contexte permettant ainsi la génération des actions de reconfigurations adéquates.

Avant d'expliquer l'approche d'analyse et de diagnostic proposée, il s'avère important d'enchaîner avec une classification des méthodes de diagnostic existantes.

### 2.3 Les méthodes de diagnostic existantes

Le diagnostic, selon la norme AFNOR, est l'identification de la cause probable de la (ou des) défaillances d'un système à l'aide d'un raisonnement logique fondé sur un ensemble d'informations provenant d'une inspection, d'un contrôle ou d'un test. Le diagnostic, alors, représente une fonction essentielle pour la sûreté, la disponibilité et la maintenabilité des systèmes. En effet, les méthodes de diagnostic existantes sont classifiées en deux catégories :

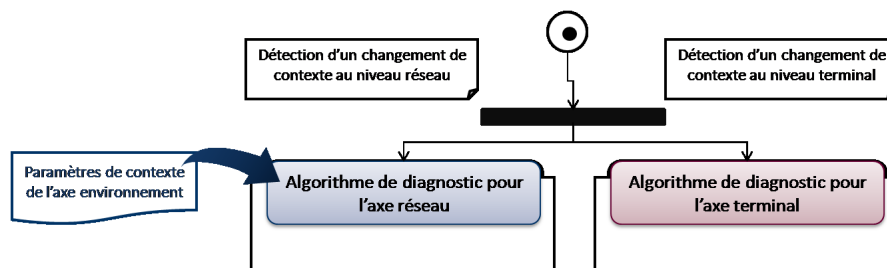


FIG. 3 – Classification des méthodes de diagnostic existantes

les méthodes de diagnostic à base de modèle et les méthodes de diagnostic guidées par les données (ou sans modèle) (voir la Figure 3) .

Le principe général des méthodes de diagnostic à base de modèle repose sur la comparaison du comportement du système réel avec le comportement du modèle établi du système. Tout écart observé entre ces deux comportements déclenche la détection d'une défaillance du système. Un modèle consiste en une reproduction du système étudié en fonctionnement nominale ou non. Les modèles, ainsi, peuvent être de nature quantitative ou qualitative.

Les méthodes de diagnostic à base de modèle peuvent être classées en deux sous-classes. La première sous-classe regroupe les méthodes de diagnostic basées sur un modèle quantitatif qui ont été développées par la Communauté FDI<sup>1</sup> (Cordier et al., 2000). Ces techniques utilisent des modèles mathématiques pour décrire le comportement nominal du système. La deuxième sous-classe renferme les méthodes de diagnostic basées sur un modèle qualitatif.

Les méthodes de diagnostic à base de modèle qualitatif sont issues de la Communauté IA<sup>2</sup> .

1. Fault detection and Isolation  
2. Communauté d'Intelligence Artificielle

Les modèles qualitatifs reposent sur des modèles symboliques afin de représenter le mode de fonctionnement des systèmes. Pour les systèmes continus, les modèles qualitatifs sont basés sur les graphes (Provan (2002)). Pour les systèmes à événements discrets, plusieurs approches sont proposées utilisant les automates ou les réseaux de Pétri (Xue et al. (2005)). Ainsi, les systèmes hybrides peuvent être modélisés à l'aide des graphes hybrides (Daigle (2008)). Bien que les méthodes de diagnostic à base de modèle possèdent plusieurs avantages. Toutefois, dans certaines applications ayant un comportement très variable au cours du temps, la tâche de modélisation du système est lourde et même difficile. En effet, les méthodes de diagnostic guidées par les données représentent une solution meilleure pour ces applications. Les méthodes de diagnostic guidées par les données reposent sur des informations issues des expériences historiques ou des résultats obtenus des expériences préalables. Parmi ces méthodes, nous citons les systèmes Expert (Zwingelstein (1995)), l'intelligence artificielle (ONDEL. (2006)) et les méthodes statistiques. Cependant, dans notre approche, il est impossible d'établir un modèle qui reflète le comportement du système à cause du dynamisme des systèmes M2M. Ainsi, nous nous référons aux méthodes de diagnostic sans modèle.

Dans la section suivante, nous présentons notre approche d'analyse et de diagnostic proposée.

### 3 Approche proposée

La conception d'une architecture M2M autonome et sensible au contexte nécessite l'implémentation de la boucle MAPE-K, introduite par IBM, permettant ainsi de collecter des données contextuelles, les analyser et déclencher les actions d'adaptation appropriées en conséquence. Dans ce contexte, nous proposons une approche d'analyse et de diagnostic distribuée, illustrée dans la Figure 4. Cette approche est destinée aux architectures M2M sensibles au contexte. Elle repose sur l'utilisation de la boucle MAPE-K d'IBM à laquelle nous proposons l'ajout d'un nouveau module de diagnostic intercalé entre la phase d'analyse et la planification. Notre approche permet, en premier lieu, de détecter les changements de contexte (phase d'analyse). En deuxième lieu, le module de diagnostic permet d'identifier les causes des changements de contexte détectés et fournir à la phase de planification les informations nécessaires afin d'exécuter les actions d'adaptation adéquates et mieux ajuster le comportement du système.

Avant de détailler notre approche proposée, nous jugeons important de mettre l'accent sur une classification des paramètres de contexte.

#### 3.1 Classification des paramètres de contexte

Nous considérons la classification des paramètres de contexte réalisée par (Chaari et al. (2005)). Cette classification comporte un vecteur à quatre dimensions :

**Contexte = (Réseau, Terminal, Utilisateur, Environnement)**

1. Réseau : cette dimension renferme les caractéristiques des réseaux de communications utilisés incluant la bande passante, la charge des liens, le taux de perte, la latence.

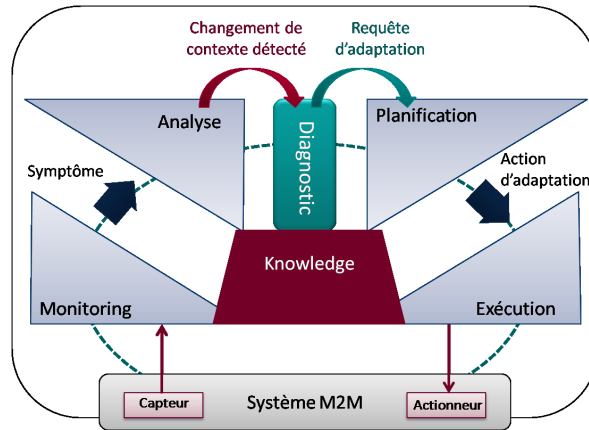


FIG. 4 – Le diagnostic dans la boucle MAPE-K

2. Terminal : cet axe comprend les caractéristiques matérielles et logicielles des terminaux utilisés comme la taille de la mémoire, la fréquence CPU, la charge CPU, le niveau de batterie et le logiciel installé.
3. Utilisateur : elle décrit toutes les caractéristiques de l'utilisateur final de l'application telles que le profil, les droits d'accès et la capacité.
4. Environnement : cette dimension enrichit la définition du contexte par d'autres paramètres selon le domaine d'applications incluant la localisation des terminaux, la vitesse de déplacement, la température, la luminosité, l'humidité.

Dans ce qui suit, nous décrivons notre approche d'analyse et de diagnostic proposée. En effet, dans cet article, l'approche proposée est relative à l'axe réseau. Dans ce contexte, nous considérons que la latence est le paramètre principale à observer.

Tout d'abord, nous rappelons notre phase d'analyse proposée dans ( Lahyani et al. (2012)).

### 3.2 Approche d'analyse proposée pour l'axe réseau

L'approche d'analyse proposée ( Lahyani et al. (2012)) se base, essentiellement, sur une comparaison des valeurs de latence par rapport à une valeur seuil adaptative. Il est à noter que le paramètre latence peut être définie comme étant : le temps écoulé entre le moment d'émission d'un message par une machine  $M1$ , noté  $t_{envoi}$  et le moment de la réception du même message par la machine  $M2$ , noté  $t_{reception}$ . Ainsi, la latence est calculée par l'Equation 1 suivante :

$$latence = t_{reception} - t_{envoi} - offset \quad (1)$$

Cependant, l'offset représente le décalage entre l'horloge de la machine  $M1$  et l'horloge de la machine  $M2$ .

En effet, les valeurs de la latence présentent souvent des fluctuations transitoires causées par la nature dynamique du réseau déclanchant un processus de recherche de routes. Pour faire face



à ces fluctuations, nous considérons l'utilisation des seuils adaptatifs. Ainsi, notre approche d'analyse pour l'axe réseau s'appuie, en premier lieu, sur le Théorème des Valeurs Extrêmes (TVE) dans le but de modéliser et étudier les valeurs maximales de la série construite à partir des valeurs de latence afin de calculer les valeurs des seuils. En deuxième lieu, l'adaptation des seuils repose sur la technique de la moyenne mobile exponentielle pondérée EWMA. Une fois calculée, nous comparons la valeur du seuil par les valeurs de la latence pour détecter les changements de contexte au niveau réseau.

Après avoir détecté un changement de contexte au niveau réseau, notre approche enchaîne par l'identification de la cause de ce changement à travers le module de diagnostic.

### 3.3 Approche de diagnostic proposée pour l'axe réseau

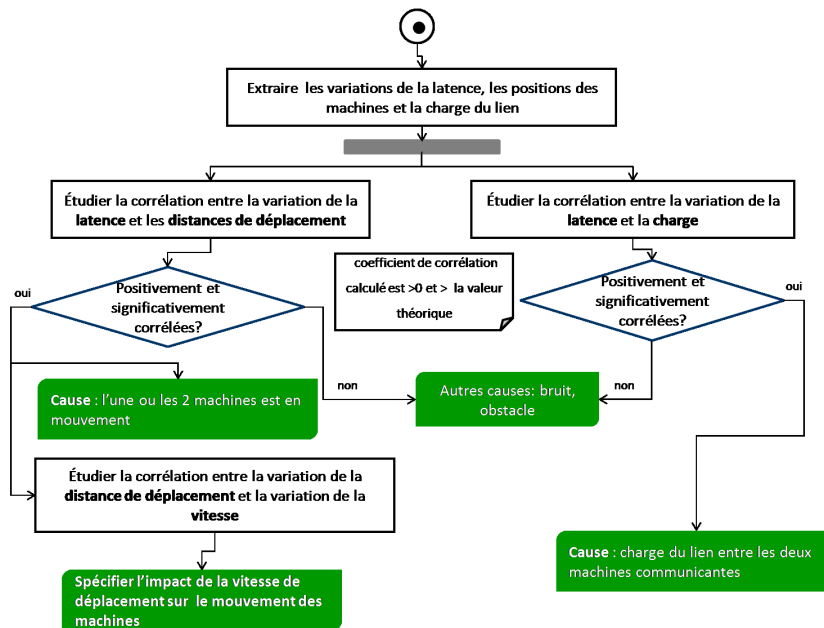


FIG. 5 – Algorithme de diagnostic pour l'axe réseau

Notre algorithme de diagnostic proposé débute après la détection d'un changement de contexte au niveau réseau. Il s'appuie sur des méthodes statistiques en se basant sur la méthode de corrélation (référence : cor pour identifier les causes des changements de contexte détectés au niveau réseau.

En effet, la méthode de corrélation permet de mesurer l'intensité de la relation qui existe entre deux distributions  $X$  et  $Y$  de même taille. Elle nécessite deux étapes. La première étape s'intéresse à calculer le coefficient de corrélation de Pearson, dénoté  $PCC$ , entre ces deux distribu-

tions à travers la formule suivante :

$$PCC(X, Y) = \frac{cov(X, Y)}{\sqrt{var(X) - var(Y)}} \quad (2)$$

Ainsi, la deuxième étape consiste à exécuter un test de significativité du coefficient de corrélation calculé en comparant cette valeur calculée avec une valeur théorique extraite à partir de la table des valeurs critiques du coefficient de corrélation de Pearson.

Pour l'axe réseau, un changement de contexte est détecté suite à une valeur de latence très élevée. L'algorithme de diagnostic de l'axe réseau, illustré dans la Figure 5, exécute en parallèle deux processus.

Le premier processus consiste à étudier la corrélation entre la variation de la latence et la variation des distances de déplacement des machines vu que l'environnement des applications traitées est mobile.

Le deuxième processus traite la corrélation entre la variation de la latence et celle de la charge puisque le surcharge des liens de transmission favorise la perte de certains messages.

Ainsi, selon la significativité des coefficients de corrélations calculés entre ces différentes distributions, nous procédons à trois scénarios différents.

- **Scénario 1 : la variation de la latence et la variation des distances de déplacement des machines sont significativement et positivement corrélées :** Nous déduisons que la mobilité peut être l'une des causes du changement de contexte détecté. Puis, pour chaque machine en mouvement, nous étudions la corrélation qui existe entre la variation de la distance de son déplacement et la variation de sa vitesse afin de spécifier l'impact de la vitesse de déplacement sur le mouvement des machines.
- **Scénario 2 : la variation de la latence et la variation de la charge sont significativement et positivement corrélées :** Nous affirmons que la charge du lien peut être l'une des causes du changement de contexte détecté. En effet, nous désignons par la charge de lien le nombre de message reçu par une machine en unité de temps.
- **Scénario 3 :** Nous déduisons que le changement de contexte est causé par des facteurs externes tels que les bruits ou les obstacles qui peuvent affecter la transmission des messages entre les machines.

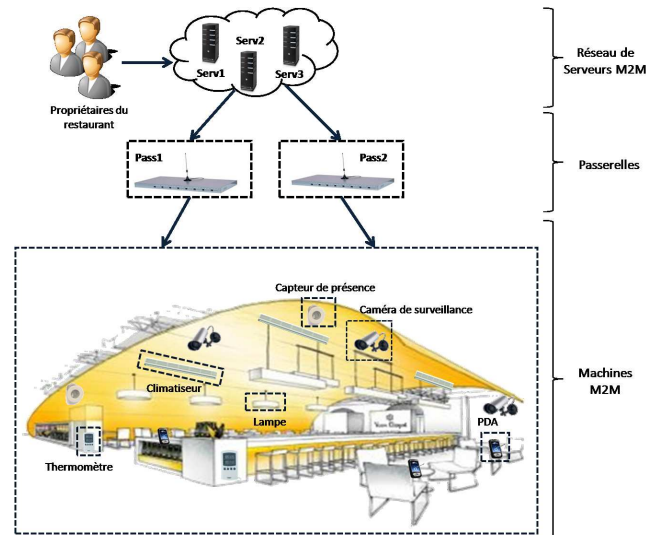
Afin d'illustrer la faisabilité de notre approche, nous présentons une étude de cas illustrative dans la section suivante.

## 4 Étude de cas

Afin d'illustrer l'utilisation de notre approche, nous présentons un exemple d'une application M2M nommée "Télé-surveillance d'un restaurant intelligent", décrite dans la Figure 6, qui vise à :

- Surveiller le matériel et la communication entre les différents équipements du restaurant,
- Assurer une bonne qualité de service à la clientèle du restaurant,
- Réduire la consommation d'énergie dans le restaurant,
- Assurer la sécurité du restaurant.

Ainsi, l'application comprend trois types de participants comme présentés dans la Figure 6 : des serveurs M2M (Serv1, Serv2 et Serv3), deux passerelles (pass1 et pass2) et différents types de machines, mentionnées dans la Table 4.

FIG. 6 – *Vue matérielle de l'application*

Les serveurs M2M permettent de visualiser et contrôler le système. Les machines M2M peuvent être des capteurs ayant pour rôle de collecter les données contextuelles ou des actionneurs jouant rôle de réagir vis-à-vis des changements de contexte.

Alors, afin d'assurer la communication entre les différents participants de l'application, il est possible de déployer un système Publier/Souscrire, comme illustré dans la Figure 7. En effet, les brokers sont déployés sur les serveurs M2M qui implémentent des algorithmes d'analyse et de diagnostic assurant l'identification des causes des changements de contexte. Une fois identifiées, les serveurs procèdent à des actions d'adaptation. Les producteurs sont déployés sur les capteurs et les compteurs afin de fournir les informations de surveillance (batterie épuisée, connexion interrompue). Les consommateurs sont déployés sur les actionneurs qui reçoivent les notifications répondant à leur besoin et réagissent en conséquence.

– **Exemple illustratif de l'approche d'analyse et de diagnostic pour l'axe réseau :**

Le propriétaire du restaurant décide de contrôler, quotidiennement, la fiabilité des liens de communications séparant les différents machines pour assurer une meilleure circulation des données au sein du restaurant.

Le propriétaire du restaurant transmet, ainsi, des messages à tous les machines du système pour vérifier la nature de l'état de connexion dans le restaurant.

Une fois le serveur détecte à l'instant  $t$  un changement de contexte (à travers le module d'analyse), l'algorithme de diagnostic débute par la collection des positions des deux machines communicantes à travers leur capteurs de position GPS intégrés.

Ensuite, l'algorithme de diagnostic extrait la charge de ce lien à partir de sa base de données. Enfin, il identifie que la mobilité du  $PDA_2$  (le PDA est attaché à un serveur du restaurant) est la cause du changement de contexte détecté en calculant la valeur du coefficient de corrélation en fonction des 3 dernières valeurs de latence et les 3 dernières valeurs de la distance de

Vers une approche d'analyse et de diagnostic pour les architectures M2M

Nom de la machine	Rôle de la machine	Type de la machine
PDA	Commander des repas (client) et gérer les commandes (employeur)	Capteur et/ou actionneur
Caméra de surveillance	Surveiller la sécurité du restaurant	Capteur et/ou actionneur
Capteur de position GPS	Déterminer la position des machines mobiles	Capteur
Capteur de présence	Capturer la présence d'une clientèle	Capteur
Compteur d'énergie	Détecter le niveau de batterie de différentes machines	Capteur
Thermomètre	Détecter la température du restaurant	Capteur
Lampe	Allumer ou éteindre la lumière	Actionneur
Climatiseur	Régler la température	Actionneur
Système d'alarme	déclencher une alarme	Actionneur

TAB. 1 – Liste des machines de l'application

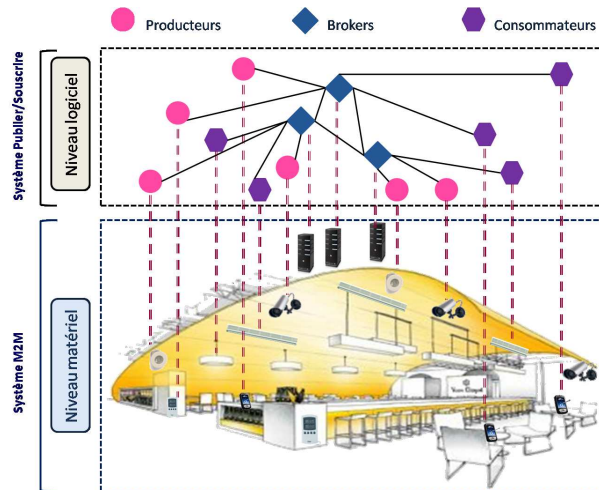


FIG. 7 – Vue logicielle de l'application

Temps de détection du changement de contexte	$t$
Variation de la latence	$latence_{t-2}$ $latence_{t-1}$ $latence_t$
Variation de la distance de déplacement	$distance_{t-2}$ $distance_{t-1}$ $distance_t$
Valeur de PCC calculée	$PCC_{cal}$
Valeur théorique du PCC	$r = 0.9877$
status du PCC calculé	$PCC_{cal} > 0$ et $PCC_{cal} > r$
<b>Cause</b>	<b>Mobilité</b> $PDA_2$ est en mouvement
Variation de la vitesse du $PDA_2$	$vitesse_{t-2}(PDA_2)$ $vitesse_{t-1}(PDA_2)$ $vitesse_t(PDA_2)$
<b>Cause</b>	$PDA_2$ se déplace avec une grande vitesse (instabilité)

TAB. 2 – Résultat de l'algorithme de diagnostic pour l'axe réseau

déplacement et en comparant la valeur du  $PCC$  calculée avec la valeur théorique. Le résultat du diagnostic est mentionné dans la Table 2.

Par conséquent, le  $PDA_2$  (c'est à dire le serveur du restaurant auquel il est attaché) sera remplacé par un autre plus stable.

Cette étude de cas justifie l'importance d'un système M2M sensible au contexte afin de faire face aux différents paramètres de contexte et d'ajuster son comportement en conséquence.

## 5 Conclusion et perspectives

Dans cet article, nous avons présenté une classification des paramètres de contexte selon quatre axes (Réseau, Terminal, Utilisateur, Environnement) afin de couvrir la totalité des situations contextuelles.

En outre, nous avons proposé une approche d'analyse et de diagnostic afin de détecter les changements de contexte qui se produisent dans les systèmes M2M et d'identifier les causes de ces changements détectés. Notre approche, ainsi, se base sur la boucle MAPE-K, introduite par IBM, et ajoute un module de diagnostic jouant l'intermédiaire entre la phase d'analyse et la phase de planification.

Le diagnostic s'appuie sur la Corrélation cherchant à identifier les causes des changements de contexte qui ont eu lieu au niveau réseau.

De plus, nous avons présenté une étude de cas intitulé “Télésurveillance d’un restaurant intelligent” illustrant la faisabilité de notre approche proposée.

Dans le futur, nous envisageons couvrir toutes les situations contextuelles en appliquant un algorithme de diagnostic pour chaque axe de la classification présentée. Puis, nous comptons mettre en place un métamodèle décrivant comment les diagnostics vont pouvoir être définis, étendus et combinés entre eux. Ensuite, nous comptons varier les scénarios de test en ajoutant de nouvelles applications plus complexes telles que la domotique et le *Smart Buildings* dans lesquelles le diagnostic peut devenir ambigu. Ces scénarios servira dans la recherche des corrélations qui peuvent être établies entre les informations de contexte appartenant à des axes différents. Enfin, nous envisageons, aussi, calculer le surcoût introduit par l’approche proposée.

## Références

<http://www.nvcc.edu/home/elanthier/methods/correlation.htm>.

- Abowd, G. D., A. K. Dey, P. J. Brown, N. Davies, M. Smith, et P. Steggle (1999). Towards a better understanding of context and context-awareness. In *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, HUC '99, London, UK, UK, pp. 304–307. Springer-Verlag.
- Alaya, M. B. et T. Monteil (2012). Frameself : A generic context-aware autonomic framework for self-management of distributed systems. In *WETICE*, pp. 60–65.
- Bouassida Rodriguez, I., G. Sancho, T. Villemur, S. Tazi, et K. Drira (2009). A model-driven adaptive approach for collaborative ubiquitous systems. In *Proceedings of the 3rd workshop on Agent-oriented software engineering challenges for ubiquitous and pervasive computing*, AUPC 09, New York, NY, USA, pp. 15–20. ACM.
- Chaari, T., F. Laforest, et A. Flory (2005). Adaptation des applications au contexte en utilisant les services web. In *Proceedings of the 2nd French-speaking conference on Mobility and ubiquity computing*, UbiMob '05, New York, NY, USA, pp. 111–118. ACM.
- Cioara, T., I. Anghel, I. Salomie, M. Dinsoreanu, G. Copil, et D. Moldovan (2010). A self-adapting algorithm for context aware systems. In *in the Proceedings of the 9th IEEE RoE-duNet International Conference*, Sibiu, Romania, pp. 374 – 379. IEEE INDEXED.
- Computing, A., W. Paper, et T. Edition (2006). An architectural blueprint for autonomic computing . *Quality* 36(June), 34.
- Cordier, M., P. Dague, M. Dumas, F. Levy, J. Montmain, M. Staroswiecki, et L. Trave-Massuyes (2000). Ai and automatic control theory approaches of model-based diagnosis : links and underlying hypotheses. In *In Proceedings of Safeprocess*, pp. 274–279.
- Daigle, M. J. (2008). *A qualitative event-based approach to fault diagnosis of hybrid systems*. Ph. D. thesis, Nashville, TN, USA. AAI3378184.
- Lahyani, I., L. B. Amor, M. Jmaiel, K. Drira, et C. Chassot (2012). Analytical framework for qos aware publish/subscribe system deployed on manet. In *Proceedings of the 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications*, ISPA '12, Washington, DC, USA, pp. 325–331. IEEE Computer Society.

- Lupu, E., N. Dulay, M. Sloman, J. Sventek, S. Heeps, S. Strowes, K. Twidle, S.-L. Keoh, et A. Schaeffer-Filho (2008). Amuse : autonomic management of ubiquitous e-health systems. *Concurr. Comput. : Pract. Exper.* 20(3), 277–295.
- Nabet, D., M. Rizcallah, et D. Kaplan (2006). Machine to machine, enjeux et perspectives.
- ONDEL., O. (2006). *Diagnostic par reconnaissance des formes : application a un ensemble convertisseur - machine asynchrone*. Ph. D. thesis, Ecole Centrale de LYON.
- Provan, G. (2002). A model-based diagnostic framework for distributed systems. In *In Proceedings of 13th International Workshop on Principles of Diagnosis*, pp. 16–22.
- Xue, F., D.-z. Zheng, et L. Yan (2005). Fault diagnosis of distributed discrete event systems using obdd. *Informatica* 16(3), 431–448.
- Zheng, D., J. Wang, W. Han, Y. Jia, et P. Zou (2006). Towards a context-aware middleware for deploying component-based applications in pervasive computing. In *Proceedings of the Fifth International Conference on Grid and Cooperative Computing, GCC '06*, Washington, DC, USA, pp. 454–457. IEEE Computer Society.
- Zwingelstein, G. (1995). *Diagnostic des défaillances. Traité des nouvelles Technologies, série Diagnostic et Maintenance*. Hermes.

## Summary

Unavoidable development of ubiquitous systems, and in particular machine-to-machine systems (M2M) causes the complexity of the management of this type of system. The design of such an autonomic and context-aware system based on architecture for self-adaptability is a promising solution. This architecture changes the behavior of the system based on user needs, available resources and the environment of the application. In this paper, we propose an approach for the analysis and diagnosis of M2M systems based on statistical methods aimed, first, to detect changes in context (analysis phase). Second, the diagnostic module can identify the causes of changes in context detected during the analysis phase and can provide to the planning phase information necessary to perform the appropriate adaptation actions.

