

QoCIM : un méta-modèle de qualité de contexte

Pierrick MARIE¹, Thierry DESPRATS¹, Sophie CHABRIDON², Michelle SIBILLA¹

¹IRIT UMR 5505 Université Paul SABATIER, 31062 TOULOUSE, France
<prenom>.<nom>@irit.fr

²Institut Mines-Télécom, CNRS UMR 5157 SAMOVAR, Télécom SudParis, 91011 Évry, France
Sophie.Chabridon@telecom-sudparis.eu

RÉSUMÉ

Cet article rapporte une étude comparative menée sur les récents travaux successifs qui ont chacun proposé une liste de critères de qualité de contexte (QoC). Il en résulte qu'aucun consensus ne peut être établi quant à la sémantique et l'exhaustivité des critères de QoC. Face à ce constat, le méta-modèle QoCIM est proposé afin de modéliser des critères de QoC hétérogènes. Cela dans le but d'offrir, aux gestionnaires de contexte et aux applications, un moyen générique, calculable et expressif quant à la manipulation et à l'exploitation de tout critère de QoC. À titre d'évaluation, QoCIM a été expérimenté pour la modélisation d'un ensemble de critères simples et composés.

ABSTRACT

In the last decade, many works aimed to propose their own quality of context (QoC) criteria list. This article relates a comparative study of these successive propositions. The result is that no consensus has been reached about the semantic and the completeness of QoC criteria. Faced with this situation, the QoCIM meta-model is proposed as a generic, computable and expressive solution to handle and to exploit any QoC criterion within context managers and context-sensitive applications. For validation purposes, QoCIM is successfully applied for modelling a set of simple and composite QoC criteria.

Keywords

quality of context, quality criterion, context management, meta-modeling, information model

1. INTRODUCTION

Le développement des technologies de l'IoT (l'extension d'Internet aux objets du monde physique), du cloud computing, du big data, et celles favorisant la mobilité vont amener au développement d'applications ubiquitaires sensibles à des informations de contexte et des situations. Ces situations sont calculées à partir de données ambiantes, de profils utilisateurs et d'informations émanant de sources hétérogènes et spatialement distribuées. Les applications sensibles au contexte se généralisent et deviennent de plus en plus courantes. Ces applications requièrent une gestion fine et efficace de la qualité des informations qu'elles reçoivent.

La pertinence de la réaction de ces applications dépend fortement de la QoC fournies. Or, selon les objectifs métiers de ces applications, certains critères de QoC apparaîtront plus importants que d'autres pour consolider leurs prises de décisions. Parfois seule la fraîcheur (l'âge de la donnée) prévaudra, parfois la précision, d'autres fois les deux. Une solution pour répondre à ce besoin est d'utiliser des gestionnaires de contexte. Les applications exploitent alors les informations fournies par ces gestionnaires. Ils prennent en charge les informations de contexte tout au long de leurs cycles de vie. Le cycle de vie d'une information de contexte commence lors de sa création par un capteur et se termine lors de sa consommation par une application sensible aux informations de contexte. Entre ces deux événements les informations de contexte peuvent être agrégées, filtrées, déduites ou transformées [2] plusieurs fois. Or, ces informations sont par nature incomplètes et imprécises [8]. Une mauvaise qualité des informations de contexte peut entraîner de mauvaises prises de décisions et des réactions non pertinentes. C'est pourquoi les gestionnaires de contexte doivent prendre en compte la QoC durant chaque phase de leur transformation. C'est ce défi auquel devront répondre les futurs gestionnaires de contexte.

Quelques solutions ont déjà été proposées. En 2007, le projet AWARENESS [15] proposait un middleware pour la gestion des informations de contexte et offrait un moyen de gérer la QoC. En 2012, le projet COSMOS [4] offrait une solution basée sur l'ingénierie dirigée par les modèles pour la conception d'applications sensibles à la QoC. Enfin, l'un des objectifs du projet INCOME [1], lancé en 2012, est de fournir un gestionnaire de contexte qui sera capable d'adresser à la fois des problèmes de vie privée et gestion de la QoC.

Notre objectif est d'offrir un moyen *générique, calculable et expressif* pour que les gestionnaires de contexte puissent

manipuler et exploiter la QoC de manière simple et efficiente. *Générique*, car notre solution doit permettre de modéliser des critères de QoC complexes et hétérogènes. *Calculable*, car pour estimer le niveau de qualité d'une information, il faut être en mesure d'effectuer des traitements et des opérations sur ces critères. Enfin, *expressive*, car les applications sensibles au contexte doivent pouvoir exprimer leurs besoins en QoC auprès de gestionnaires de contexte différents. La section 2 compare les listes de critères de QoC qui ont été proposées lors de précédents travaux durant la dernière décennie. Après avoir constaté qu'il n'existe pas de listes standards de critères pour mesurer la QoC, la section 3 analyse différents modèles susceptibles de fournir une solution générique, calculable et expressive pour manipuler et exploiter la QoC. La section 4 décrit le méta-modèle que nous proposons et la section 5 illustre ses premières expérimentations.

2. ANALYSE COMPARATIVE DES PROPOSITIONS DE CRITÈRES DE QOC

Dans le but de fournir une solution pour modéliser la QoC, nous nous sommes intéressés aux travaux existants traitant de la mesure de la QoC. Celle-ci se base sur une liste de critères. De nombreux auteurs ont déjà établi leur propre liste de critères pour mesurer la QoC. Les paragraphes suivants énumèrent les principales propositions faites durant la dernière décennie. Enfin, le dernier paragraphe de cette section montre les résultats de notre étude comparative des critères proposés en fonction de leur sémantique. L'étude montre les divergences qui existent au niveau de la dénomination et de la signification des différents critères. En fonction des auteurs, une même signification peut être associée à plusieurs dénominations différentes. À l'inverse, pour une même dénomination des auteurs différents peuvent y associer des significations différentes.

Buchholz 2003

En 2003, BUCHHOLZ et SCHIFFERS [3] proposèrent la première liste de critères définissant la QoC. Cette liste était composée de cinq critères : «*precision*», «*probability of correctness*», «*trust-worthiness*», «*resolution*» et «*up-to-dateness*». Chacun de ces critères est défini à l'aide d'une description textuelle. Aucune méthode de calcul n'est formalisée quant à leurs estimations.

Kim 2006

KIM et LEE [10] proposaient, en 2006, une nouvelle liste de critères pour mesurer la QoC. Cette liste était construite en confrontant les critères de qualité proposés par [3] avec des critères génériques de mesure de la qualité. KIM et LEE fournissaient cinq critères avec une définition orientée du point de vue des utilisateurs finaux, consommateurs d'informations de contexte. Les critères proposés étaient «*accuracy*», «*completeness*», «*representation consistency*», «*access security*» et «*up-to-dateness*». La suite des travaux de KIM et LEE fournit une formule mathématique pour estimer la valeur de leurs deux premiers critères : «*accuracy*» et «*completeness*».

Sheikh 2007

Dans le cadre du développement du middleware AWARENESS, SHEIKH *et al.* [15] formulèrent en 2007 leur propre liste de critères de QoC. Ces critères sont «*precision*», «*freshness*», «*temporal resolution*», «*spatial resolution*», et «*probability of correctness*». Bien que les critères soient longuement décrits de manière

textuelle, aucune méthode n'est formulée pour estimer leurs valeurs. Les descriptions des critères de QoC adoptent à la fois le point de vue du producteur et du consommateur des informations de contexte. Ici, les producteurs sont les entités qui créent et fournissent les informations de contexte, comme des capteurs. Tandis que les consommateurs sont les applications sensibles aux informations de contexte.

Filho 2010

En 2010, après avoir analysé les critères de QoC proposés par différents auteurs auparavant, dont ceux de [3], de [10] et [15], FILHO [7] proposa une nouvelle liste de critères de QoC. Il redéfinit les critères «*up-to-dateness*», «*sensitiveness*», «*access security*», «*completeness*», «*precision*» et «*resolution*». Il fournit une formule mathématique ou un algorithme pour estimer chacun de ces critères.

Neisse 2012

NEISSE [12] suggéra, en 2012, d'adopter le standard ISO utilisé en métrologie pour définir les critères de QoC. Il établit que les concepts de «*accuracy*» et «*precision*» utilisés comme critères n'étaient qu'une définition approximative du critère de précision utilisé en métrologie. Il en va de même pour les concepts de «*spatial resolution*» et «*temporal resolution*» définis par [15] qui ne sont pour NEISSE qu'une redéfinition du standard ISO de la précision sur des informations spatiales et temporelles. NEISSE suggéra alors de mesurer la QoC avec deux critères : l'âge et la précision. L'âge correspond à l'intervalle de temps écoulé depuis la production d'une information de contexte. Le critère de précision applique le standard ISO de la précision sur d'autres types d'informations en fonction des besoins d'une application. Ainsi, ce critère peut être appliqué à la localisation de la source de l'information de contexte par exemple.

Manzoor 2012

La liste la plus complète de critères de QoC fut publiée en 2012 par MANZOOR, TRUONG, et DUSTDAR [11]. Ils établissent sept critères de qualité de haut niveau, chacun dépendant d'autres critères pour être calculé. Pour chacun de ces critères de haut niveau, une formule mathématique est fournie afin de les estimer. Les critères proposés sont «*reliability*», «*timeliness*», «*completeness*», «*significance*», «*usability*», «*access right*», «*representation consistency*». La définition de certains critères adopte le point de vue du producteur de l'information de contexte, comme la précision d'une information fournie par un capteur par exemple. Tandis que la définition d'autres critères adopte le point de vue des applications sensibles aux informations de contexte comme la fraîcheur maximale acceptée.

Après avoir étudié la sémantique des critères listés ci-dessus nous avons constaté des divergences. La même dénomination d'un critère apparaît dans plusieurs listes proposées par différents auteurs sans être associée à la même signification. Inversement, des significations différentes sont associées à une même dénomination de critère dans des listes différentes. Il existe également des significations associées à des dénominations qui ne sont présentes que dans une seule liste. Ces divergences sont illustrées dans la figure 1. La figure regroupe les critères étudiés par auteur et montre les divergences les plus significatives qui existent entre tous ces critères. La figure synthétise les différentes dénominations et

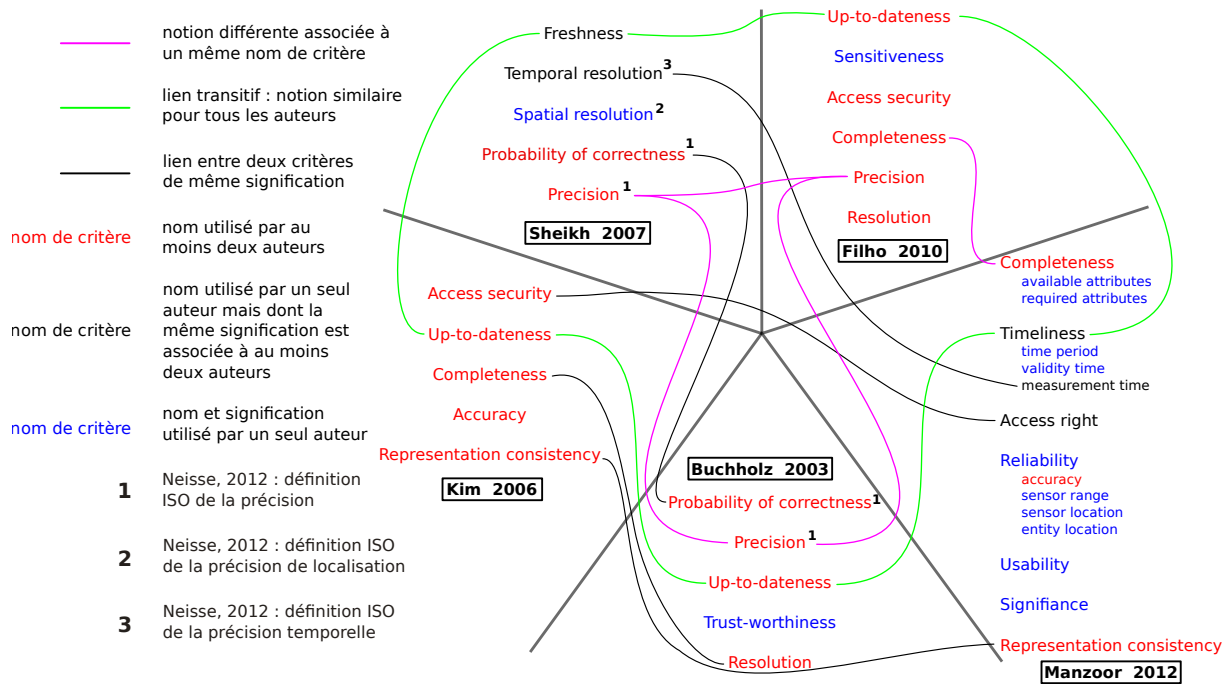


Figure 1. Comparaison des critères de QoC de différents auteurs

significations qui existent entre les critères étudiés. Le lien en vert sur la figure met également en avant la seule notion qui semble commune entre toutes les listes proposées, la notion de fraîcheur. Avec cette figure, nous constatons qu'il n'existe donc pas de consensus sur les critères de qualité à utiliser pour mesurer la QoC. Cette constatation appuie les propos de [2] en 2012 indiquant qu'un consensus sur la définition de critères de QoC complexes restait un problème ouvert.

3. ÉTUDES DE MODÈLES EXISTANTS

Notre objectif est de fournir une solution générique, calculable et expressive. Puisqu'il n'existe pas de consensus sur la liste de critères à utiliser pour mesurer la QoC, l'aspect générique de notre solution ne peut pas être basé sur une liste unique et exhaustive de critères. Il faut donc fournir un moyen générique capable d'intégrer différentes listes de critères de QoC dont les dénominations, les significations et les méthodes de calcul divergent. Une solution pour répondre à ce besoin de généricité est d'utiliser un modèle ou un méta-modèle de qualité de contexte. C'est pourquoi plusieurs modèles ont été étudiés. Ces modèles sont utilisés dans différents domaines proches ou similaires à la gestion de contexte. D'autres sont des modèles standardisés ou dédiés qui pourraient apporter des concepts pertinents pour écrire des modèles de critères de QoC. Le premier modèle étudié est celui de l'Open Geospatial Consortium (OGC)¹ [14] de 2007 qui représente des observations faites principalement par des capteurs. Il associe au résultat d'une observation le processus qui a été suivi pour le produire. Il associe également au résultat un niveau de qualité représenté par une chaîne de caractères. Or, la mesure de la QoC se base sur une liste de critères. Un critère étant au minimum composé d'une dénomination, d'une signification et d'une valeur, utiliser une chaîne de caractères comme le fait le modèle de l'OGC

n'est pas suffisant notamment en terme de calculabilité. C'est pourquoi il n'a pas été retenu. Le second modèle étudié est le méta-modèle d'information du projet IoT-A² [9]. Ce projet a pour but de fournir des méthodes et des solutions de conception génériques permettant de créer des applications reposant sur l'Internet des Objets. Il apporte la notion de méta-données associées à une information. Avec cette notion, il est possible d'assimiler les méta-données à des critères de qualité et ainsi de répondre à notre besoin d'une solution générique. Cependant, cette interprétation nécessite beaucoup d'effort d'enrichissement du méta-modèle avant de pouvoir l'exploiter correctement. C'est pourquoi ce méta-modèle n'a pas été retenu et notre étude s'est poursuivie vers d'autres modèles capables de fournir une solution plus générique, calculable et expressive. Les trois modèles suivants, respectivement proposés par le projet COSMOS, le Distributed Management Task Force (DMTF)³ et l'Object Management Group (OMG)⁴, se sont avérés plus pertinents pour répondre à notre objectif. Les paragraphes reportent l'analyse de ces modèles.

Le méta-modèle de COSMOS

COSMOS est un gestionnaire de contexte utilisé pour des applications consommatrices d'informations de contexte. Il utilise un système de contrats pour gérer les niveaux de QoC reçus par les applications. Ce méta-modèle [5], objet de l'annexe 1, est utilisé pour concevoir des contrats de gestion de la QoC. Un contrat, *ContextAwarenessContract*, est établi entre une application consommatrice d'informations de contexte et une source fournissant des informations de contexte. Un contrat définit le niveau de qualité requis, *QoCLevel*, pour une application. Ces niveaux sont définis pour des paramètres

¹www.opengeospatial.org

²www.iiot-a.eu

³www.dmtf.org

⁴www.omg.org

de qualité, *QoCParameter*, d'une information de contexte. Le méta-modèle offre trois types de contrats *ObservationContract*, *NotificationContract* et *AdaptationContract*. Une application choisit la méthode qui lui convient le mieux pour recevoir les informations de contexte. Le méta-modèle utilisé dans COSMOS gère très finement les contrats de qualité établis entre les consommateurs et les producteurs d'informations de contexte. Ce modèle, à base de contrats, répond donc bien à l'aspect expressif de la solution que nous recherchons. Cependant, l'aspect calculable de la solution que nous cherchons n'est pas totalement couvert par ce méta-modèle. En effet, les critères de QoC sont représentés par la classe *QoCParameter* qui ne contient aucun attribut. Les propriétés d'un critère ne sont donc pas clairement établies ce qui empêche donc d'avoir une solution générique et calculable.

Le modèle CIM du DMTF

Le document [6] du DMTF modélise des métriques CIM (Common Information Model). Ce modèle, voir l'annexe 2, est l'un des standards développés par le DMTF. Il modélise et contrôle des métriques sur des éléments gérés. Les éléments gérés sont principalement des ressources de l'information et de la communication (TIC). Nous analyserons ce modèle en considérant les informations de contexte comme des éléments gérés. De cette manière, les métriques CIM sont assimilables à des critères de QoC. En effet, la classe abstraite *ManagedElement* peut donc représenter l'information de contexte à qualifier. La classe *BaseMetricDefinition* représente un critère de qualité. La classe *BaseMetricValue* représente alors une estimation de ce critère. Les classes héritées de *BaseMetricDefinition* et *BaseMetricValue* conçoivent des critères de qualité plus complexes. Le modèle utilise plusieurs patrons de calcul de métriques comme l'agrégation ou la sélection de valeurs discrètes. Pour les aspects de généralité et de calculabilité de la solution que nous recherchons, la technique utilisée par CIM est intéressante. Elle sépare la définition d'une métrique de sa valeur. Elle permet de mesurer la QoC à l'aide d'une liste de critères dont les significations et les méthodes de calcul divergent. Chacun de ces critères est ensuite associé à des estimations de sa valeur. En revanche, ce modèle ne couvre pas l'aspect expressif de la solution que nous recherchons. De plus, les métriques CIM sont initialement conçues pour superviser des TIC, bien qu'une interprétation du modèle est possible pour la supervision d'informations de contexte, le coût d'adaptation du modèle CIM pour la gestion de listes de critères de QoC hétérogènes semble élevé.

Le méta-modèle de QoS de l'OMG

Le dernier modèle étudié [13], voir l'annexe 3, est fourni par l'Object Management Group. Il ne traite pas directement de la QoC, mais de la qualité de service (QoS). L'étude de ce modèle permet de mettre en avant d'éventuels points communs entre la gestion de la QoS et celle de la QoC. C'est la raison pour laquelle nous avons étudié ce méta-modèle. À l'image des métriques CIM, l'OMG sépare la partie définition d'un critère de sa valeur grâce aux classes *QoSDimension* et *QoSValue*. Il regroupe les définitions des critères dans des catégories afin de mieux les manipuler grâce à la classe *QoSCharacteristic*. La classe *QoSDimension* possède plusieurs attributs dont *direction* qui compare deux valeurs d'un même critère du point de vue des utilisateurs du service qualifié. Par exemple, une plus grande valeur du critère du temps de réponse d'un service signifie, pour les utilisateurs de ce service, qu'il est moins intéressant. Il est

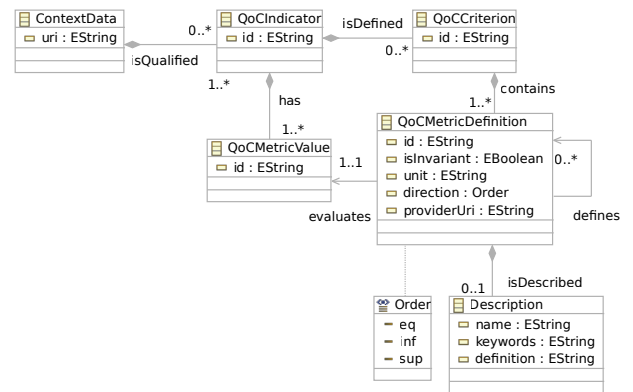


Figure 2. QoCIM : QoC Information Model

également possible de spécifier avec l'attribut *unit* quelle est l'unité, dans le système international par exemple, de la valeur d'un critère. L'attribut *statisticalQualifier* décrit quelle est la méthode statistique utilisée pour fournir une valeur du critère de qualité. La classe *QoSConstraint* est une superclasse utilisée pour établir des contrats de QoS. L'attribut *qualification* de cette classe spécifie la manière de réaliser le contrat, «*best-effort*» ou «*guarantee*» par exemple. L'OMG modélise la QoS à l'aide de critères. Ils sont ensuite utilisés dans des contrats de gestion de la QoS. Le méta-modèle de l'OMG regroupe donc la notion de contrats de qualité, présente dans le méta-modèle utilisé par COSMOS, et la notion de définitions de critères, présente dans le modèle CIM. Ce méta-modèle fédère ainsi en quelques classes les deux notions qui répondent à notre attente d'une solution générique, calculable et expressive. Nous n'avons pas repris en l'état le méta-modèle de l'OMG, cependant, le méta-modèle que nous proposons, *QoCIM* (QoC Information Model), dans la section suivante en est fortement inspiré.

4. QOCIM : UN NOUVEAU MÉTA-MODÈLE DE QOC

Le méta-modèle que nous proposons, *QoCIM*, en figure 2, offre un support de modélisation générique de critères de QoC. Son cœur est composé des classes *QoCIndicator*, *QoCCriterion*, *QoCMetricValue*, *QoCMetricDefinition* et *Description*. Une information de contexte, *ContextData*, est qualifiée par des indicateurs, *QoCIndicator*. Un indicateur contient au moins une valeur, *QoCMetricValue*. Il regroupe, par exemple, un historique de plusieurs valeurs d'un même critère de qualité, ou des valeurs de plusieurs critères de qualité similaires calculées de différentes manières. Un indicateur est également défini par des critères de qualité, *QoCCriterion*. La classe *QoCCriterion* regroupe différentes *QoCMetricDefinitions* de même nature en fonction du type de valeur, *QoCMetricValue*, qu'elles produisent. Ces valeurs sont calculées à partir de la classe *QoCMetricDefinition*. Cette classe définit les méthodes d'évaluation des critères de QoC. Ses attributs définissent la production de *QoCMetricValue*. *isInvariant* indique si la valeur produite par une *QoCMetricDefinition* est une constante qui n'est pas modifiable ou si elle est calculée dynamiquement. *unit* représente l'unité de la valeur produite, généralement ce sont des unités du système international. *providerUri* identifie la ressource qui fournit la *QoCMetricValue*. Cet attribut permet de filtrer la QoC en fonction de l'entité qui l'a calculée. *direction* compare différentes *QoCMetricValues* entre elles du point de

vue consommateur des informations de contexte. Les valeurs possibles de cet attribut sont *inf*, *sup* et *eq*. *inf* signifie qu'avec une plus grande *QoCMetricValue*, le niveau de QoC est moins bon. Par exemple, la fraîcheur d'une information est calculée régulièrement avec la formule suivante : *date courante - date production de l'information*. Plus le temps passe, plus le résultat de ce calcul augmente tandis que la qualité de l'information diminue. *sup* signifie qu'avec une plus grande *QoCMetricValue*, le niveau de QoC est meilleur. Par exemple, le critère d'exactitude d'une information est évalué à partir de la probabilité que cette information soit correcte. Plus cette probabilité augmente, plus la qualité de l'information augmente. *eq* signifie que différentes *QoCMetricValue* ne modifient pas directement le niveau QoC perçu par le consommateur, par exemple, le statut d'un capteur. La classe *Description* apporte de la sémantique à une *QoCMetricDefinition*. Le nom de la description est contenu dans l'attribut *name*, *keywords* contient une liste de mots clés et *definition* est un texte décrivant la *QoCMetricDefinition*. Comme dans le modèle CIM, *QoCIM* sépare la définition d'une métrique, *QoCMetricDefinition*, de sa valeur, *QoCMetricValue*. Par rapport au méta-modèle de l'OMG, *QoCIM* reprend quelques-uns de ses attributs comme *isInvariant*, *direction* et *unit*. Le modèle CIM et le méta-modèle de l'OMG permettent à leurs manières de construire des définitions de métriques à partir d'autres métriques. *QoCIM* le permet également grâce au lien récursif au niveau de la classe *QoCMetricDefinition*. À cela nous avons ajouté l'attribut *providerUri* ainsi que la classe *Description* qui ne sont pas présents dans le méta-modèle de l'OMG. La section suivante montre une expérimentation de *QoCIM*.

5. EXPÉRIMENTATIONS

Nous avons expérimenté *QoCIM* avec les critères définis par Manzoor et al. [11]. Nous avons choisi cette liste de critères car elle est la liste la plus riche et la plus aboutie que nous ayons étudiée. Manzoor et al. ont défini vingt et un critères de QoC qu'ils ont classifié en quatre catégories, à savoir «*sensor characteristics*», «*measurement context*», «*specifications and consumer requirements*» et «*QoC metric*». Le tableau 1 du document [11] résume les critères proposés. Les trois premières catégories contiennent des critères simples car ils ne dépendent pas d'autres critères pour être estimés. Par exemple, la mesure de l'intervalle de temps entre deux mesures faites par un capteur ne dépend pas d'autres critères. La dernière catégorie contient des critères composés. Ils dépendent des critères définis dans les trois premières catégories pour être calculés. Par exemple, la fraîcheur d'une information est calculée à partir de sa date de création et de l'intervalle de temps entre deux mesures de cette information. Ce sont ces critères composés qui ont été cités dans la section 2 et sont utilisés dans la figure 1.

Nous avons expérimenté *QoCIM* à l'aide des technologies EMF⁵ et GMF⁶ basées sur Eclipse. L'expérimentation s'est déroulée en trois étapes. La première consistait à modéliser *QoCIM* à l'aide de la technologie EMF. La seconde étape était d'utiliser la modélisation EMF de *QoCIM* pour représenter les critères simples de [11]. La dernière étape consistait à représenter, avec la modélisation EMF de *QoCIM*, les critères composés de [11] en se

⁵Eclipse Modeling Framework
www.eclipse.org/modeling/emf

⁶Graphical Modeling Project
www.eclipse.org/modeling/gmp

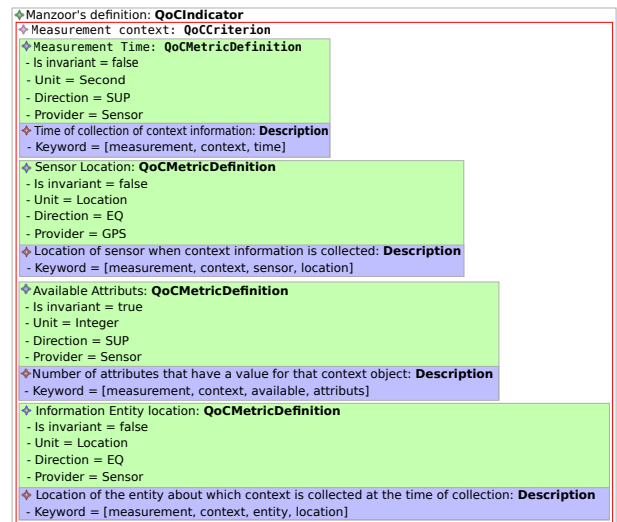


Figure 3. Critères de mesure définis par MANZOOR

basant sur les représentations des critères simples produits lors de la seconde étape.

Étape 1 : le méta-modèle EMF

Cette étape consistait à construire un méta-modèle EMF de *QoCIM*. Elle a permis de vérifier que *QoCIM* était valide syntaxiquement. La figure 2 provient de cette modélisation.

Étape 2 : représentation de critères simples

La seconde étape reposait sur l'utilisation de la technologie GMF pour produire un éditeur capable de manipuler des instances de notre méta-modèle EMF. La technologie GMF fournit une «*tool-chain*» qui prend en entrée un méta-modèle EMF et produit en sortie un éditeur d'instances de ce méta-modèle. Une fois la «*tool-chain*» GMF correctement configurée, un éditeur d'instances de *QoCIM* est produit. Nous avons ensuite utilisé cet éditeur pour modéliser les critères simples de Manzoor et al. La figure 3 représente la modélisation des critères de la catégorie «*measurement context*». Le cadre rouge représente un *QoCCriterion*. Les rectangles verts correspondent aux *QoCMetricDefinition* et les rectangles violets aux *Description*. Les autres critères des catégories «*sensor characteristics*» et «*specifications and consumer requirements*» ont également été modélisés mais ne sont pas représentés dans ce document. Cette étape nous a permis de vérifier que *QoCIM* peut modéliser des critères simples de QoC.

Étape 3 : représentation de critères composés

La dernière étape consistait à vérifier que *QoCIM* est capable de modéliser les critères composés de Manzoor et al. La modélisation de ces critères est basée sur la modélisation des critères simples effectuée dans l'étape précédente. Nous avons repris les critères simples produits précédemment et les avons combiné en fonction des définitions des critères composés proposées par Manzoor et al. Pour ce faire, nous avons utilisé la «*tool-chain*» GMF comme dans la seconde étape. La figure 4 montre une partie des critères composés ainsi modélisés. La figure contient les critères qui dépendent d'un grand nombre de critères simples. La figure représente en rose les critères composés de [11]. Les autres couleurs représentent

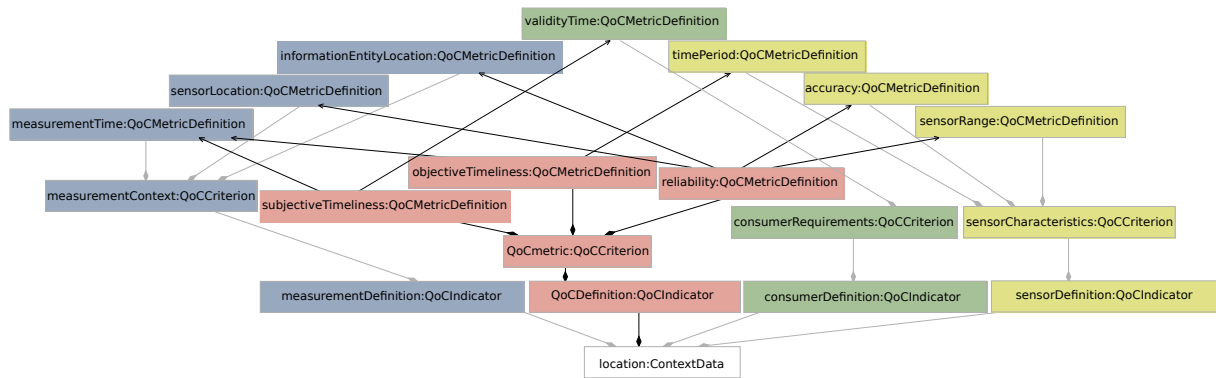


Figure 4. Une instantiation de QoCIM appliquée aux critères de MANZOOR

les critères simples modélisés à l'étape précédente. En bleu, sur le côté gauche de la figure, se trouvent les critères de la catégorie «*measurement context*», ceux représentés dans la figure 3. En vert, à droite, se trouvent les critères de la catégorie «*specifications and consumer requirements*». Enfin, en jaune à l'extrême droite de la figure, se trouvent les critères de la catégorie «*sensor characteristics*». La figure 4 démontre que la modélisation de critères composés est possible avec QoCIM.

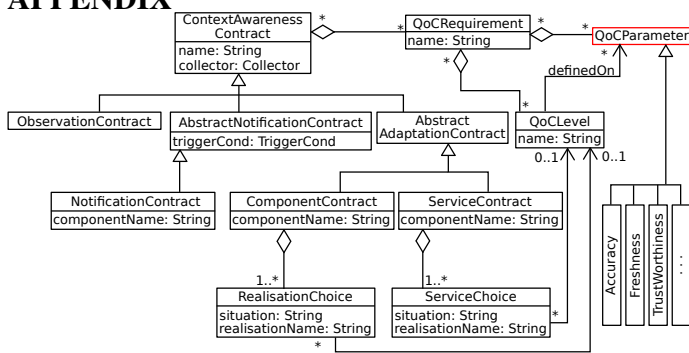
6. CONCLUSION

Durant la dernière décennie, des travaux se focalisant sur la gestion de la QoC ont été menés. Nous avons présenté dans cet article le résultat de l'étude empirique que nous avons menée dans le but de comparer les listes de critères successivement proposées par différents auteurs. De façon explicite, nous avons montré les divergences existantes et conclu sur la difficulté d'atteindre un consensus quant à l'établissement d'une liste unique et exhaustive de critères de QoC. Face à ce constat, et afin de manipuler au sein des gestionnaires de contexte tout critère de QoC souhaité, nous avons conçu QoCIM, un méta-modèle dédié à la manipulation de la QoC construit à partir de concepts pertinents issus de plusieurs modèles dédiés ou standardisés pour d'autres domaines. Dans cet article, nous avons présenté le cœur informationnel de QoCIM et avons éprouvé avec succès sa capacité d'instanciation pour modéliser la liste la plus riche et la plus aboutie des critères de QoC proposée très récemment par MANZOOR *et al.* Nous travaillons actuellement à l'extension de QoCIM pour intégrer de nouveaux concepts aptes à modéliser en particulier l'expression d'exigences pouvant porter sur des critères et sur des niveaux de QoC. Ceci afin d'offrir un mécanisme permettant aux producteurs d'informations de contexte de spécifier la QoC qu'ils fournissent et aux consommateurs d'informations de contexte de spécifier la QoC qu'ils demandent. Cela permettra aussi aux gestionnaires de contexte d'évaluer la QoC tout le long de la chaîne de traitement des informations de contexte et d'appliquer des politiques de filtrage basé sur la QoC.

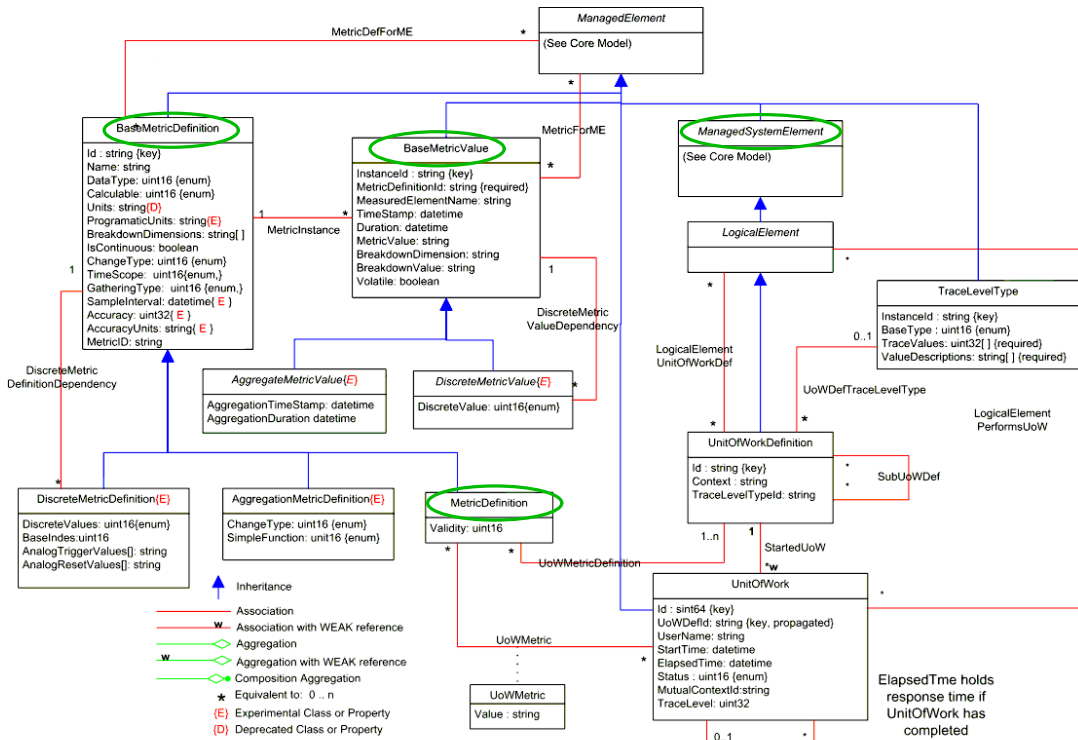
7. REFERENCES

- [1] J.-P. Arcangeli, A. Bouzeghoub, V. Capms, M.-F. Canut, S. Chabridon, D. Conan, T. Desprats, R. Laborde, S. Leriche, H. Maurel, A. Peninou, C. Taconet, and P. Zaraté. Projet INCOME: INfrastructure de gestion de CONtexte Multi-Échelle pour l'Internet des Objets. In *Ubimob '12*, 2012.
- [2] P. Bellavista, A. Corradi, M. Fanelli, and L. Foschini. A Survey of Context Data Distribution for Mobile Ubiquitous Systems. 2012.
- [3] T. Buchholz and M. Schiffers. Quality of context: What it is and why we need it. In *In Proceedings of the 10th Workshop of the OpenView University Association*, 2003.
- [4] S. Chabridon, D. Conan, Z. Abid, and C. Taconet. Ingénierie dirigée par les modèles pour la construction d'applications ubiquitaires sensibles à la qualité du contexte. In *Ubimob '12*, 2012.
- [5] S. Chabridon, Z. Conan, Denis. and Abid, and C. Taconet. Building ubiquitous qoc-aware applications through model-driven software engineering. *Science of Computer Programming*, 2012.
- [6] Distributed Management Task Force. *Base Metric Profile*, 2009.
- [7] J. d. r. Filho. *A Family of Context-Based Access Control Models for Pervasive Environments*. PhD thesis, University of Grenoble Joseph Fourier, 2010.
- [8] K. Henriksen and J. Indulska. Modelling and using Imperfect Context Information. In *Proc. 1st PerCom Workshop CoMoRea*, 2004.
- [9] Internet of Thing Architecture. *Deliverable 1.3*, 2012.
- [10] Y. Kim and K. Lee. A quality measurement method of context information in ubiquitous environments. In *Proceedings of the International Conference on Hybrid Information Technology*, 2006.
- [11] A. Manzoor, H.-L. Truong, and S. Dustdar. Models and applications for context-aware systems in pervasive environments. *Knowledge Engineering Review Special Issue on Web and Mobile Information Services*, 2012.
- [12] R. Neisse. *Trust and Privacy Management Support for Context-Aware Service Platforms*. PhD thesis, University of Twente, Enschede, 2012.
- [13] Object Management Group. *Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms Specification*, 2008.
- [14] O.G.C. *Observations and Measurements – Part 1*, 2007.
- [15] K. Sheikh, M. Wegdam, and M. Van Sinderen. Middleware support for quality of context in pervasive context-aware systems. In *Pervasive Computing and Communications Workshops*, 2007.

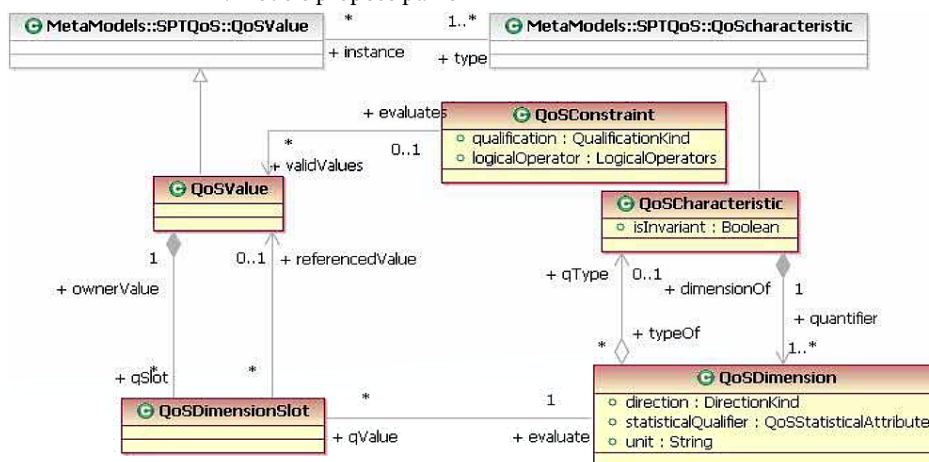
APPENDIX



ANNEXE 1. Méta-modèle utilisé dans COSMOS



ANNEXE 2. Modèle proposé par le DMTF



ANNEXE 3. Méta-modèle proposé par l'OMG