

Gestionnaire multiagent de contexte pour les applications d'intelligence ambiante

A. Sorici^{†,*}, G. Picard[†], O. Boissier[†], A. Florea^{*}

[†]Laboratoire Hubert Curien UMR CNRS 5516, Institut Henri Fayol, MINES
Saint-Etienne, France

^{*}University Politehnica of Bucharest, Department of Computer Science, Romania

29 juin 2015

financement partiel par

Sectoral Operational Programme Human Resources Development 2007-2013 du
Ministère roumain des fonds européens par l'accord de financement
POSDRU/159/1.5/S/134398

Outline

Introduction

CONCERT

Flexibilité

Utilisation et Expérimentation

Conclusions - Perspectives

Outline

Introduction

CONCERT

Flexibilité

Utilisation et Expérimentation

Conclusions - Perspectives

Contexte



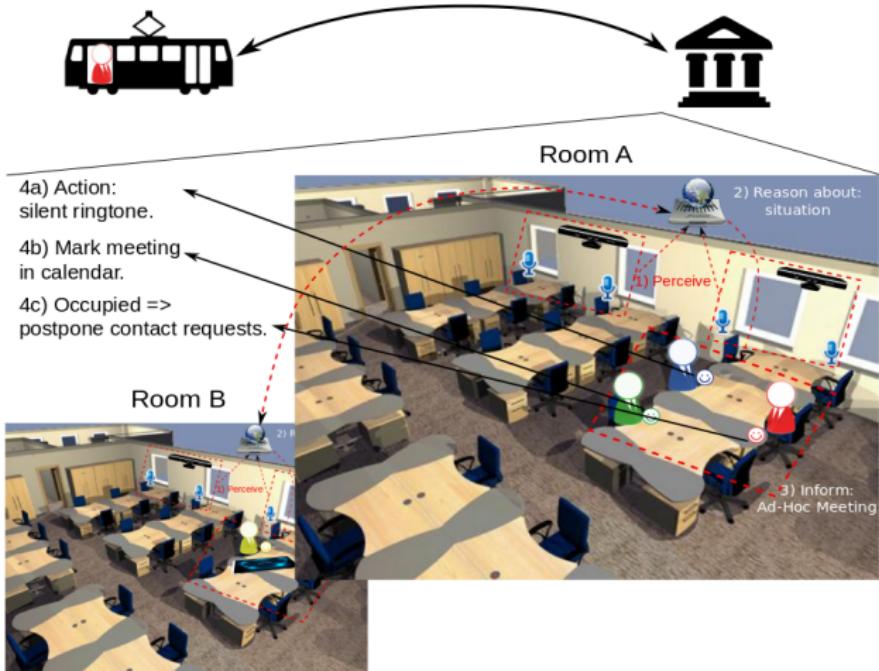
Contexte



Contexte



Contexte



Défis et objectif

Défis

- ▶ Large palette de cas d'application
- ▶ Grande variabilité des scénarios d'usage, des applications et des capteurs
- ▶ Incertitude et Informations incomplètes
- ▶ Diversité des cycles de provisionnement du contexte avec de multiples focus d'intérêt

Objectif

- ▶ Proposer un intergiciel de gestion de contexte capable d'aborder cette diversité au moment du développement et de l'exécution
 - Modélisation expressive du Contexte
 - Provisionnement flexible du Contexte
 - Gestion et déploiement flexible du gestionnaire de contexte

Outline

Introduction

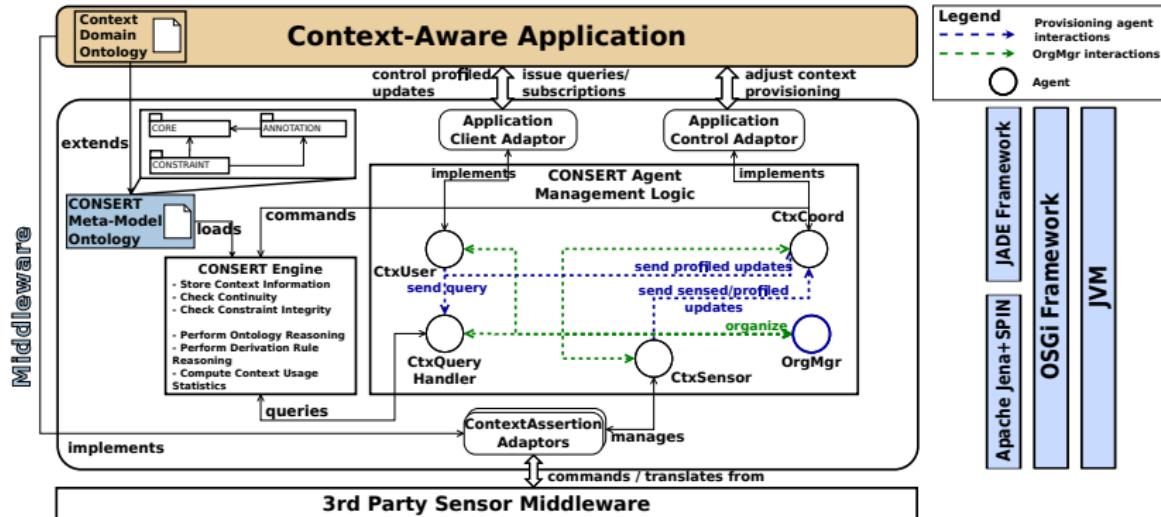
CONCERT

Flexibilité

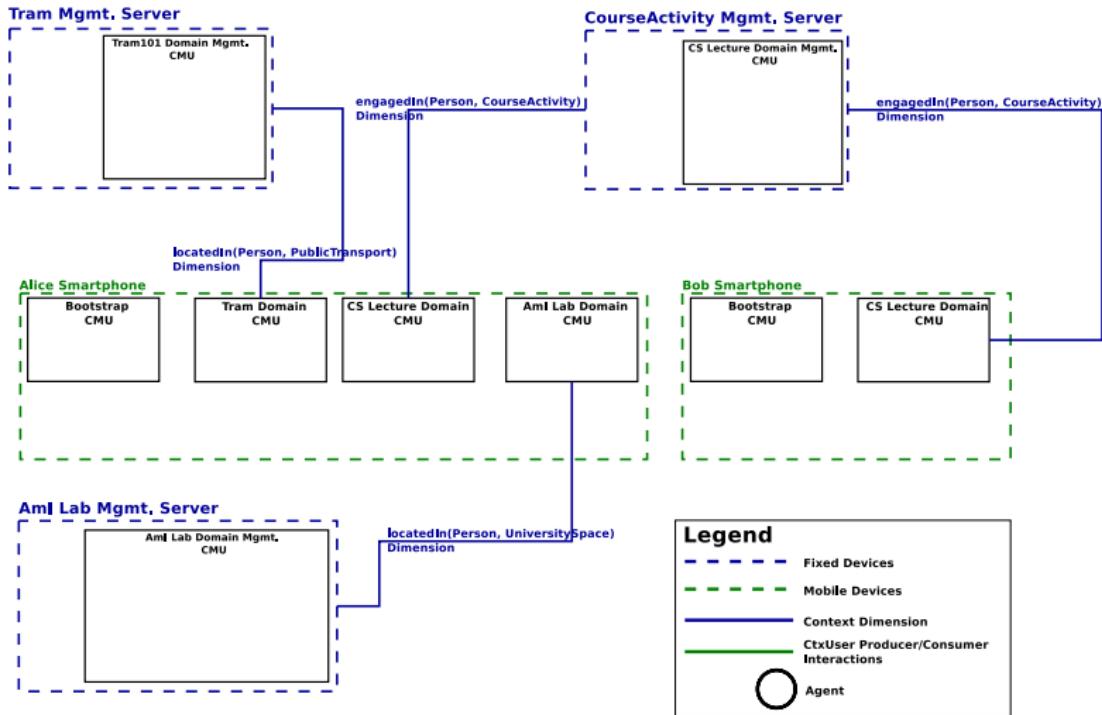
Utilisation et Expérimentation

Conclusions - Perspectives

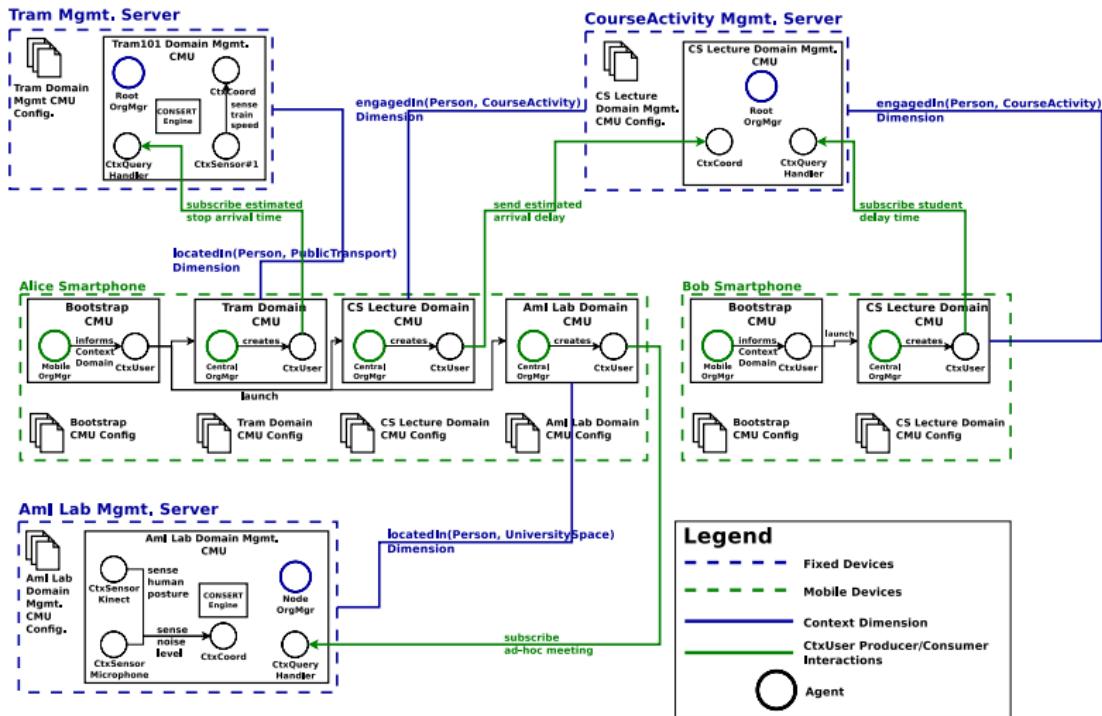
Vision globale: Context Management Unit



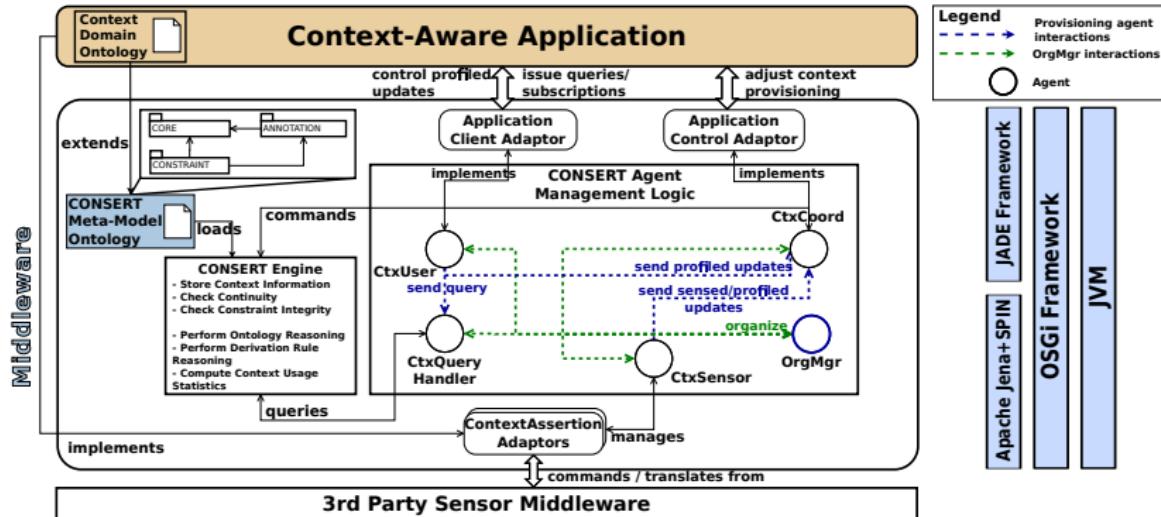
Vision globale: CONCERT Context Management Middleware



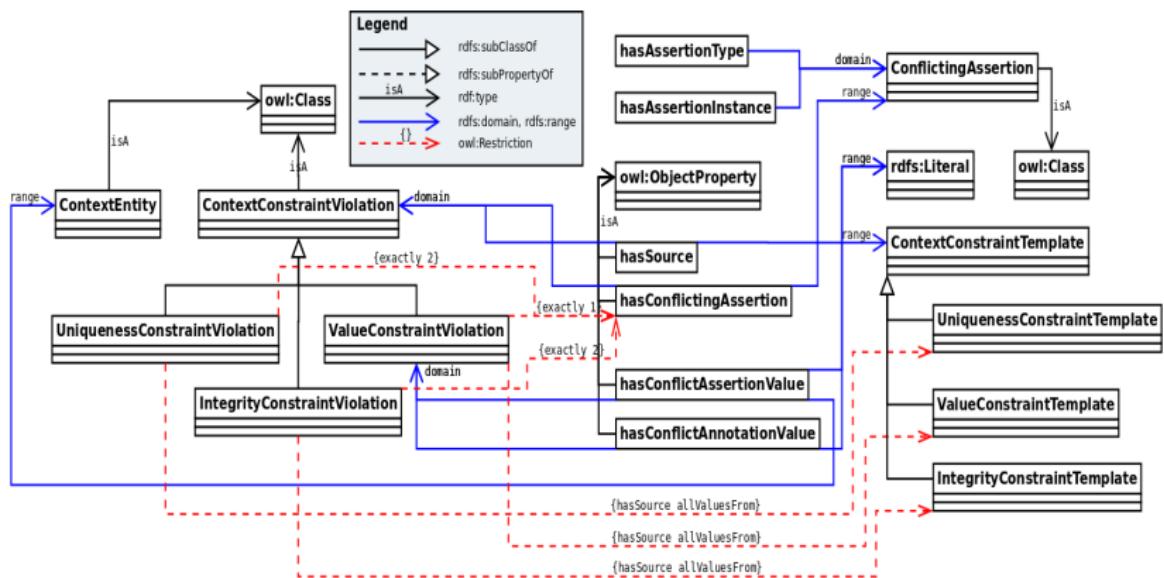
Vision globale: CONCERT Context Management Middleware



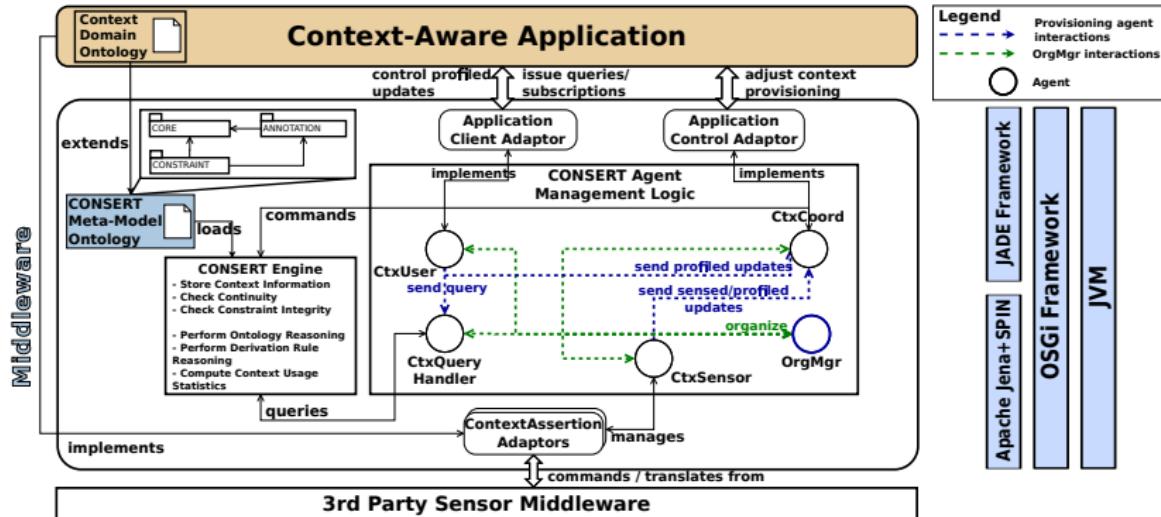
Vision globale: Modélisation du Contexte



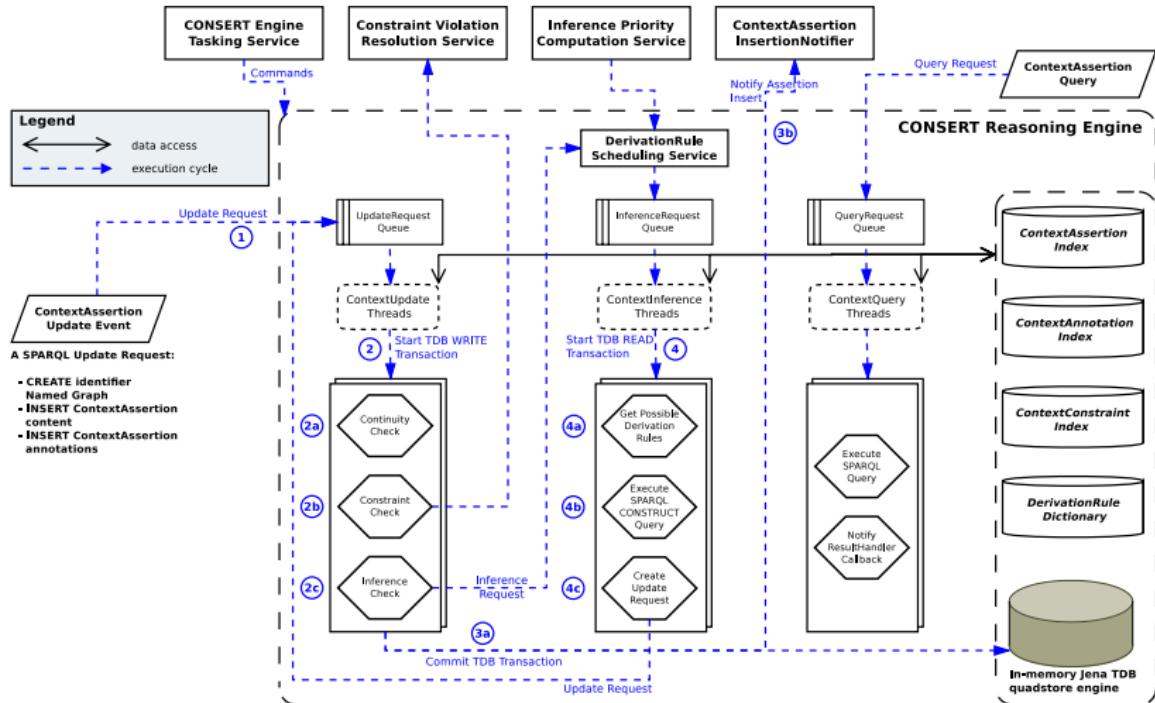
Vision globale: Modélisation du Contexte



Vision globale: Raisonnement sur le Contexte



Vision globale: Raisonnement sur le Contexte



Outline

Introduction

CONCERT

Flexibilité

Utilisation et Expérimentation

Conclusions - Perspectives

Flexibilité du déploiement

- ▶ Organisation et structuration des CMU selon les caractéristiques "Dimension" (e.g. spatiale, activité) et "Domaine" du contexte de l'application
- ▶ Styles de déploiement : Centralisé, Décentralisé
- ▶ Politiques de déploiement adressant :
 - configuration de la plateforme support (e.g. container JADE)
 - configuration des domaines de contexte
 - configuration et répartition des agents des CMU
- ▶ Vocabulaire de spécification du déploiement représenté dans ontologie

Flexibilité du provisionnement de contexte

Intra-CMU

- ▶ Politiques de provisionnement de contexte: paramètres et règles de contrôle
 - Politiques de perception (Sensing Policies)
 - Politiques de coordination

Inter-CMU

- ▶ Protocoles de provisionnement de contexte
 - Chaine de perception (CtxCoord et CtxSensor agents)
 - Chaine de requête: queries/subscriptions

Outline

Introduction

CONCERT

Flexibilité

Utilisation et Expérimentation

Conclusions - Perspectives

Implémentation

- ▶ Modèle de contexte: représentation et moteur de Raisonnement
 - Ontologies publiées
 - RDF Named Graphs
 - Apache Jena
 - SPIN (SPARQL Inferencing Notation)
- ▶ Agents de provisionnement de contexte
 - Framework JADE
- ▶ Middleware :
 - Java
 - OSGi

Evaluation

Qualitative

- ▶ Flexibilité de l'ontologie CONCERT pour modéliser les scénarios de référence
- ▶ Modélisation et résolution de dépendances entre contraintes de contexte
- ▶ Ecriture et exécution de règles de dérivation de contexte
- ▶ Influence des politiques de provisionnement de contexte
- ▶ Capacité à spécifier et déployer facilement différents types de scénarios

Quantitative

- ▶ Performance du moteur de Raisonnement
- ▶ Performance des requêtes distribuées entre les agents de provisionnement

Outline

Introduction

CONCERT

Flexibilité

Utilisation et Expérimentation

Conclusions - Perspectives

Conclusions & Perspectives

Contributions

- ▶ Méta-modèles de contexte fondé sur les ontologies (représentation uniforme), Moteur de raisonnement configurable
- ▶ Système multi-agent pour les Contexte Management Units
- ▶ Adaptation et flexibilité du déploiement et du provisionnement (politiques déclaratives, protocoles d'interaction, schéma de déploiement)
- ▶ Implémentation, Evaluation, Test

Perspectives

- ▶ Enrichir le méta modèle avec des annotations
- ▶ Améliorer les performances du moteur
- ▶ Introduire la notion de Context Level Agreement pour gérer l'adaptation et la flexibilité