

AsaWoO : exemple d'utilisation de ROS pour le Web des objets

Workshop ROS - INNOROBO – 21/03/2013

Michaël Mrissa – Lionel Médini – Thibaud Houel – Tanguy Raynaud-Gallonet

Laboratoire d'InfoRmatique en Image et Systèmes d'information

LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/Ecole Centrale de Lyon

<http://liris.cnrs.fr>



Plan de la présentation

☰ Présentation du projet

- Consortium
- Verrous scientifiques
- Objectifs opérationnels

☰ Architecture globale

☰ État d'avancement

☰ Intégration avec ROS

☰ Conclusion

Le projet AsAWoO

Projet soumis à l'ANR INFRA 2013

- En cours d'évaluation...

Consortium

- LIRIS, LIP (Université Lyon 1)
- LCIS Valence (Université de Grenoble)
- Génération Robots (Bordeaux)
- IRISA (Université de Vannes)

Le projet AsAWoO

Contexte historique

● Internet et le Web

● Pile de protocoles « stable »

- *TCP/IP puis HTTP,*

● Technologies Web en évolution rapide

- *AJAX, Web Sockets...*

● Les objets

● Pile de protocoles adhoc

● Équipements hétérogènes

Le projet AsAWoO

Verrous / domaines scientifiques

- **Intégration des objets dans le Web**
 - Architectures orientées services
- **Augmentation des capacités des objets**
 - Intelligence artificielle
 - Context-aware computing
- **Composition de fonctionnalités avancées**
 - Systèmes Multi-Agents
- **Optimisation des communications**
 - Green IT & Green by IT
- **Déploiement, réplication, scalabilité**
 - Cloud computing
- **Aspects sécurité et confidentialité**

Le projet AsAWoO

☰ Objectif global du projet

- Intégration naturelle des objets dans le Web

☰ Objectifs technologiques

- Mise en œuvre de fonctionnalités complexes
- Fédération d'objets
- Interaction à travers le Web

☰ Objectifs opérationnels

- Définition / standardisation de « WoT applications »
- Distribution par des « WoT marketplaces »

Plan de la présentation

☰ Présentation du projet

☰ **Architecture globale**

- Notion d'avatar

- Schéma global

☰ État d'avancement

☰ Intégration avec ROS

☰ Conclusion

Architecture

☰ Notion d'Avatar

- Abstraction de l'objet sur le Web
- Objet réel = incarnation de l'avatar

☰ Avantages

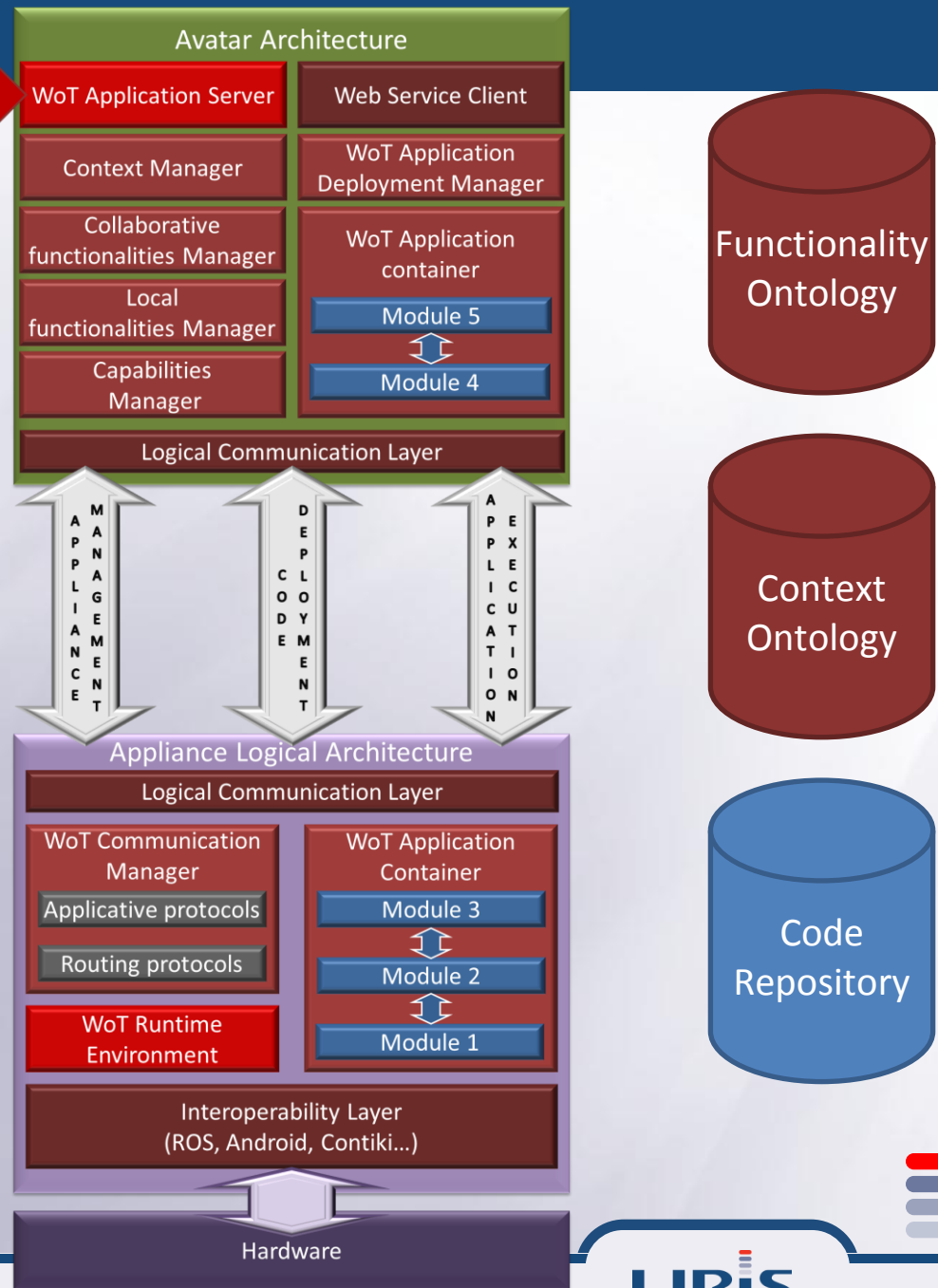
- Communication inter-avatars normalisée
- Offre de fonctionnalités en tant que services Web
 - Les avatars forment un SMA
 - Composition de services
- Déploiement à distance de code

☰ Couche d'interopérabilité

- Entre l'objet et son avatar
- Solution explorée : ROS

Architecture

Architecture globale



Plan de la présentation

- ≡ Présentation du projet
- ≡ Architecture globale
- ≡ **État d'avancement**
- ≡ Intégration avec ROS
- ≡ Conclusion

État d'avancement

Cadre

- Projets de recherche de 1^{ère} année de master informatique de l'Université Lyon 1
- Depuis Janvier 2013
- Groupes de deux étudiants

État d'avancement

Côté avatar

- Architecture de l'avatar
 - Modules internes
 - Communication inter-avatars
- Fonctionnalités
 - Aspects sémantiques

Liaison avatar-objet

- Sécurité
 - Communications avatar-objet (SSL)

État d'avancement

Côté robot

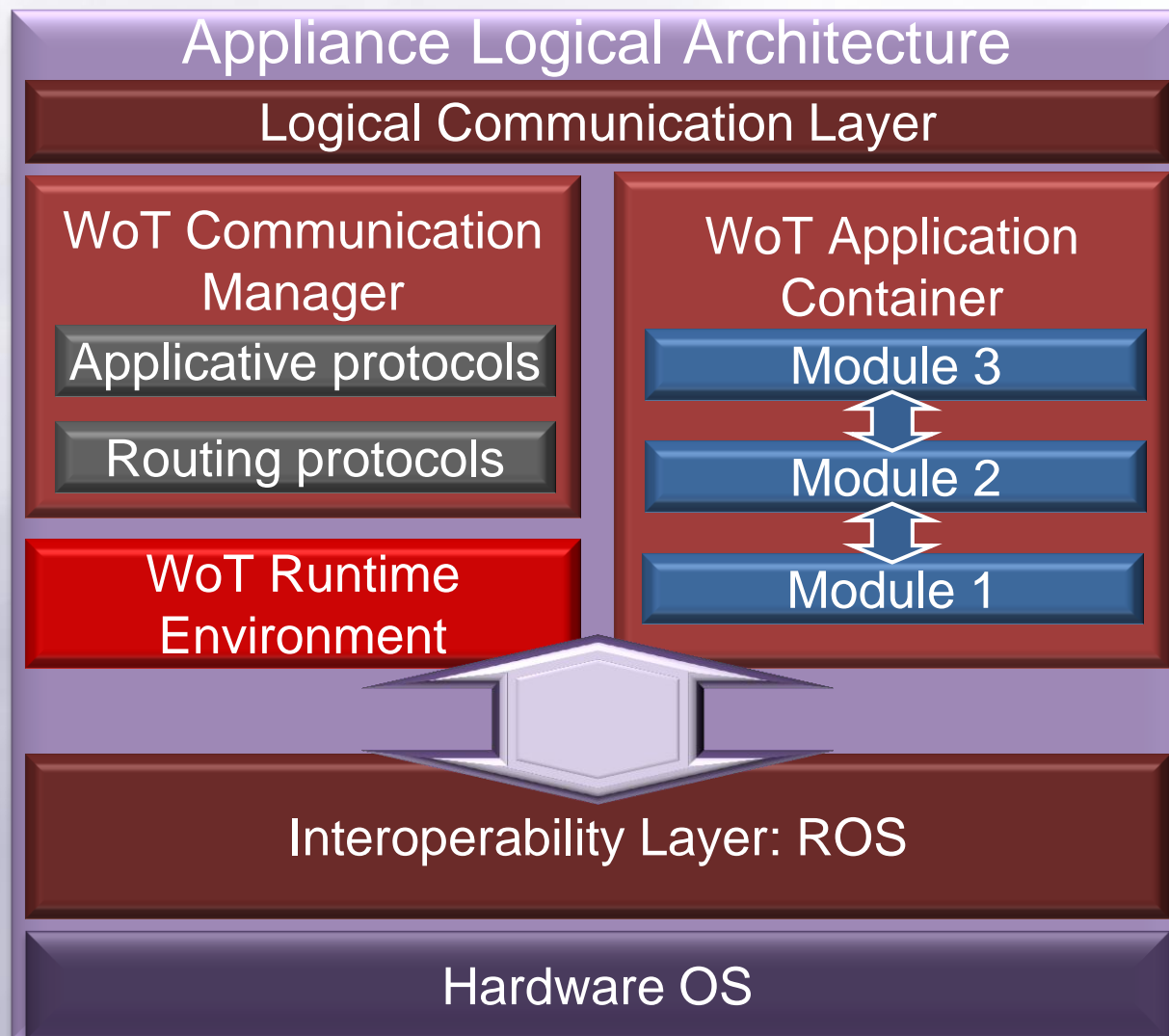
- Test de faisabilité
 - Brique NXT + LeJOS
 - Contrôle de fonctionnalités dans un client Web
- Prototype simplifié de l'architecture WoT
 - Découplage fonctionnalités / capacités
 - *Exposition sur le Web d'une fonctionnalité basique*
 - *Mise en œuvre côté serveur (Java)*
 - Ajout d'une couche d'interopérabilité
 - **ROS**
 - Test sur deux robots différents

Plan de la présentation

- ≡ Présentation du projet
- ≡ Architecture globale
- ≡ État d'avancement
- ≡ **Intégration avec ROS**
 - Architecture côté objet
 - Aspects techniques
 - Démonstration
- ≡ Conclusion

Architecture côté objet

- ☰ **Couche applicative**
 - Java serveur
- ☰ **Couche interopérabilité**
 - ROS
- ☰ **Couche physique**
 - OS de l'objet



Aspects techniques

☰ Machine hôte

- Core 2 duo, 4Go RAM, Windows 7
- Java 7 / Tomcat 7

☰ VM

- Ubuntu 11.10 « Oneiric Ocelot »
- Player : VirtualBox

☰ ROS

- Electric Emys
- Rosbridge 2.0
- NXT-ROS

☰ Robots

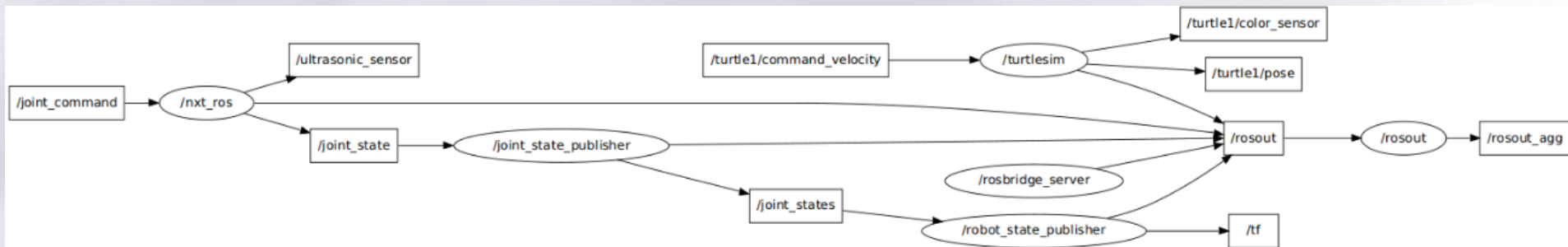
- TurtleSim
- Lego MindStorm NXT (USB)

Démonstration

☰ Fonctionnalité implémentée : « Explore »

- Avance jusqu'au prochain obstacle
- Tourne à droite

☰ Graphe des objets



☰ Démonstration...

Plan de la présentation

- ≡ Présentation du projet
- ≡ Architecture globale
- ≡ État d'avancement
- ≡ Intégration avec ROS
- ≡ **Conclusion**
 - Retour d'expérience
 - Discussion

Retour d'expérience

ROS

● Utilité

- Modèle générique de l'objet : URDF / SRDF
- Interopérabilité entre les OS
- Beaucoup de bibliothèques disponibles

● Spécificités

- Accès sous forme de graphe
 - *Nodes, Topics (publish/subscribe), Services*
- Déploiement sur la machine cliente

● Questions en suspens

- Compatibilité des versions / bibliothèques
- Installation / déploiement de code sur l'objet

Retour d'expérience

Compétences acquises / nécessaires

● Techniques

- Langages : C++, Python
- Communication : XML-RPC, JSON, WebSockets
- Administration : Ubuntu

● ROS

- Vision globale de l'architecture
- Structuration des modules

● Générales

- Conception logicielle
- Adaptation, asservissement, programmation événementielle
- Interaction entre les couches architecturales

Merci pour votre attention

Discussion

- D'autres retours d'expérience ?
- Des questions ?

