

Une architecture orientée avatar pour le Web des Objets

Journées SEmba - Usages des systèmes embarqués –04/04/2013
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 / École Centrale de Lyon

<http://liris.cnrs.fr/>



Plan de la présentation

Présentation du projet

- Consortium
- Contexte
- État de l'art
- Objectifs opérationnels
- Verrous scientifiques

Architecture globale

Architecture embarquée

Démonstration

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

● Piles de protocoles *ad hoc*

● Équipements hétérogènes

Le projet AsAWoO

Contexte institutionnel

● Européen

- PPP « Robotics2020 »
- Annoncé en 2012, soumissions début 2013

● National

- Plan « France Robot Initiative »
- Annoncé le 19 mars 2013

● Régional

- Action spécifique WoT (projet Web Intelligence), 2011
- École d'été WoT
 - www.web-intelligence-rhone-alpes.org/Ecoles/2013/
 - 2 au 6 septembre 2013, Valence (Drôme)



Le projet AsAWoO

☰ État de l'art (1/2)

● WoT

- Webinos (FP7 ICT), BETaaS (FP7 STREP), WoO (ITEA2), SITAC (ITEA)

● Cloud robotics

- FI-WARE (Future Internet PPP program), RoboEarth (FP7)

● Modélisation / adaptation contextuelle

- FI-WARE, INCOME (ANR 2011)

● Web Services

- DPWS : SIRENA (ITEA), SODA (ITEA), SOCRADES (FP6)
- Amigo (IST)

Le projet AsAWoO

☰ État de l'art (2/2)

- **Sémantique**
 - KnowRob, OpenRobots Ontology
- **Interopérabilité**
 - ANSO (ITEA)
- **Green IT, Green-by-IT**
 - GEODES (ITEA), Ctrl-GREEN (ANR 2011), INFRA JVM (ANR 2011)
- **User experience**
 - Do-it-Yourself Smart Experiences

Le projet AsAWoO

Objectif global du projet

- Intégration naturelle des objets dans le Web

Objectifs technologiques

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

Objectifs opérationnels

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

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**

Plan de la présentation

- ≡ Présentation du projet
- ≡ **Architecture globale**
 - Notion d'avatar
 - Schéma global
- ≡ Architecture embarquée
- ≡ Démonstration
- ≡ Conclusion

Architecture

☰ Notion d'Avatar

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

☰ Avantages

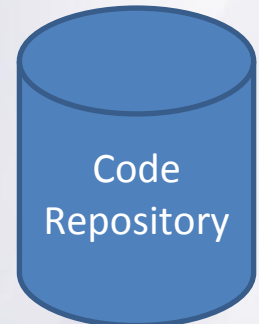
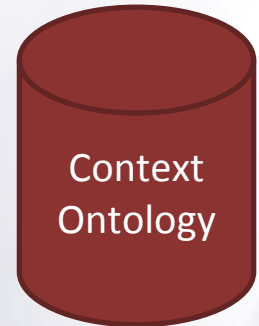
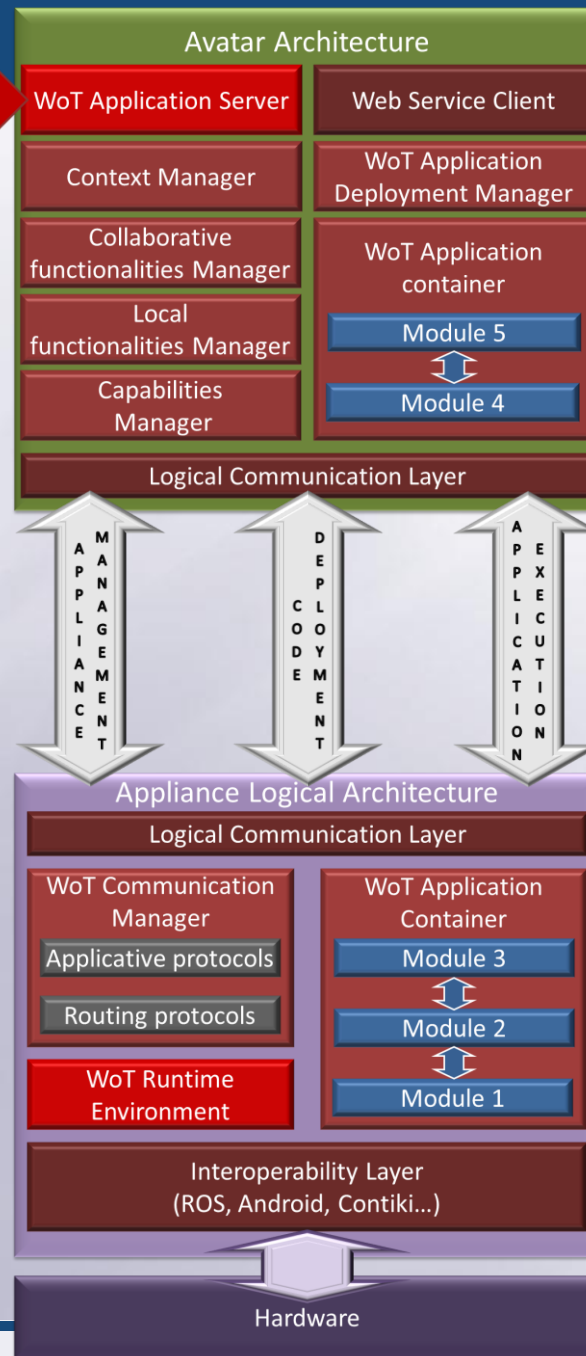
- Communication inter-avatars normalisée
- Offre de fonctionnalités en tant que services Web
 - Les avatars forment un SMA
 - Composition de services
- Déploiement à distance de code

☰ Couche d'interopérabilité

- Entre l'objet et son avatar
- Deux solutions explorées

Architecture

Architecture globale



Plan de la présentation

- ≡ Présentation du projet
- ≡ Architecture globale
- ≡ **Architecture embarquée**
 - Schéma général
 - leJOS
 - ROS
- ≡ Démonstration
- ≡ Conclusion

Architecture embarquée

☰ Couche applicative

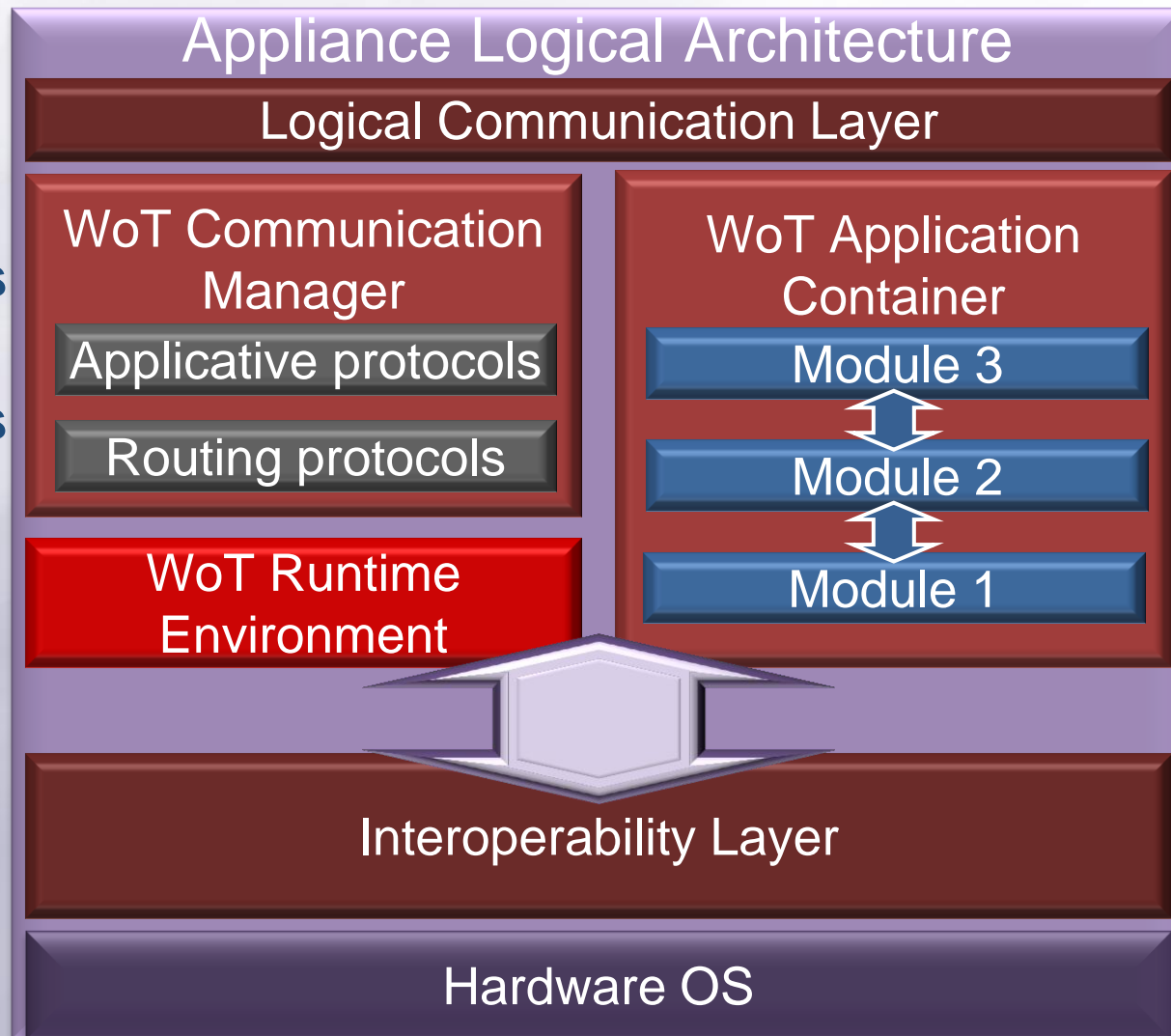
- Gestion des communications
- Gestion des applications

☰ Couche interopérabilité

- *Ad hoc / ROS*

☰ Couche physique

- OS de l'objet



Architecture embarquée

☰ leJOS (<http://lejos.sourceforge.net/>)

- OS dédié sur brique Nxt
- Programmation Java côté objet

☰ Architecture

- Communications avatar-objet
 - Protocoles
 - *Packet IO mode (sockets TCP)*
 - *Protocole applicatif : ad hoc*
 - Fonctions de base leJOS
 - Interprétation côté objet
- Côté objet
 - Serveur de capteurs / actionneurs
 - Parsing des commandes de l'avatar
 - Exécution des capacités de l'objet

Architecture embarquée

☰ ROS (<http://www.ros.org/>)

- Bibliothèque de fonctionnalités
 - Modélisation / mise à disposition des composants d'un objet
 - Interopérabilité entre les objets
- NON déployable sur brique Nxt
(connexion USB/BT avec la machine hôte)
- APIs natives : C++, Python

☰ Architecture

- Communications avatar-objet
 - ROSBridge : WebSockets + JSON
- Côté objet
 - Serveur de nœuds ROS
 - Publish / Subscribe vers des topics

Plan de la présentation

- ≡ Présentation du projet
- ≡ Architecture globale
- ≡ Architecture embarquée
- ≡ **Démonstration**
 - État d'avancement
 - Aspects techniques
 - Démonstration
- ≡ 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 SE (robot)

- **Prototype simplifié de l'architecture WoT**
 - **Couche Web**
 - *Exposition sur le Web d'une fonctionnalité basique*
 - **Couche métier (avatar)**
 - *Découplage fonctionnalités / capacités*
 - **Couche d'interopérabilité**
 - *Manuel / ROS*
 - **Couche communication**
 - *BlueTooth*
 - *Protocoles applicatifs : ad hoc / WebSockets*
 - **Couche physique (objet)**
 - *Brique NXT + LeJOS*
 - *NXT + ROS*
 - *ROS TurtleSim*

Aspects techniques

☰ Machine hôte

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

☰ leJOS

- Version
 - leJOS NXJ 0.9.1 beta
- Robot
 - Lego MindStorm NXT (BlueTooth)

Aspects techniques

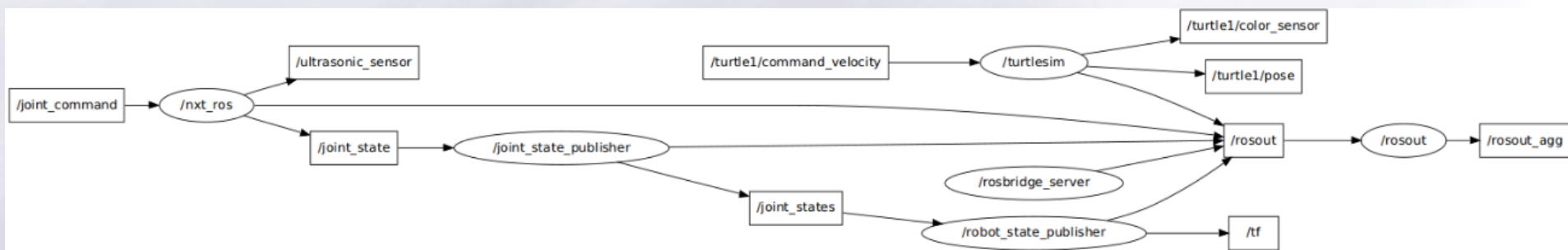
ROS

- VM
 - Ubuntu 11.10 « Oneiric Ocelot »
 - Player : VirtualBox
- Versions
 - Electric Emys
 - Rosbridge 2.0
 - NXT-ROS
- Robots
 - TurtleSim
 - Lego MindStorm NXT (USB)

Aspects techniques

ROS

● Graphe d'objets (URDF)



Démonstration

☰ Fonctionnalité implémentée : « Explore »

- Avance jusqu'au prochain obstacle
- Tourne à droite

☰ Démonstration...

Plan de la présentation

- ≡ Présentation du projet
- ≡ Architecture globale
- ≡ Architecture embarquée
- ≡ Démonstration
- ≡ **Conclusion**
 - Retour d'expérience
 - Perspectives

Retour d'expérience

leJOS

- **Utilité**
 - Interopérabilité avatar-objet (Java)
 - Effectivement implanté sur l'objet
- **Limites**
 - Solution spécifique Lego™
 - Java Micro Edition
- **Questions en suspens**
 - Modélisation / découverte des capacités de l'objet
 - Compatibilité du code Java
 - Installation / déploiement de code sur l'objet (OSGi)

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

Perspectives

☰ Ajout de sémantique

- Modélisation de l'objet
- Gestion des fonctionnalités
- Adaptation contextuelle

☰ Aspects dynamiques

- Découverte de capacités (au sens large)
- Déploiement de code
- Protocoles applicatifs

☰ Interopérabilité

- Test sur d'autres OS embarqués
→ Android ?

☰ Ajout de fonctionnalités



Merci pour votre attention

 Des questions ?

