



Lámpsakos

E-ISSN: 2145-4086

lampsakos@amigo.edu.co

Fundación Universitaria Luis Amigó

Colombia

Guzmán-Luna, Jaime; Meza González, Juan; Castro Hincapíe, Eimer
SISTEMA DE MANIPULACIÓN REMOTA DE ROBOTS ARDUINO POR MEDIO DE
SERVICIOS WEB SEMÁNTICOS

Lámpsakos, núm. 7, enero-junio, 2012, pp. 85-93

Fundación Universitaria Luis Amigó
Medellín, Colombia

Disponible en: <https://www.redalyc.org/articulo.oa?id=613965334006>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en [redalyc.org](https://www.redalyc.org)

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

SISTEMA DE MANIPULACIÓN REMOTA DE ROBOTS ARDUINO POR MEDIO DE SERVICIOS WEB SEMÁNTICOS

REMOTE MANIPULATION SYSTEM OF ARDUINO ROBOTS THROUGH SEMANTIC WEB SERVICES

Ph.D. Jaime Guzmán-Luna

Universidad Nacional de Colombia

Sede Medellín, Colombia

jaguzman@unal.edu.co

Ing. Juan Meza González

Universidad Nacional de Colombia

Sede Medellín, Colombia

judmezago@unal.edu.co

Ing. Eimer Castro Hincapíe

Universidad Nacional de Colombia

Sede Medellín, Colombia

ewcastroh@unal.edu.co

(Artículo de **Investigación científica y tecnológica**. Recibido el 31/10/2011. Aprobado el 19/12/2011)

Resumen. Este artículo presenta la implementación de un sistema basado en servicios web semánticos, orientados hacia la manipulación remota de robots capaces de ejecutar jugadas básicas de fútbol, diseñados y desarrollados con la plataforma de hardware libre Arduino. Dichos servicios fueron descritos semánticamente con el lenguaje ontológico OWL-S y bajo una ontología de dominio común que describe un robot capaz de jugar fútbol.

Palabras clave: Robótica; Robótica en la nube; Servicios web; Web semántica.

Abstract. This paper presents an implementation of a system based on different semantics web services, aimed at manipulating robots capable of executing basic plays soccer, designed and developed using the open hardware platform Arduino. These services were described semantically using ontology language OWL-S and under a common domain ontology to describe a robot that can play soccer.

Keywords: Robotics, cloud robotics, web services, semantic web

1. INTRODUCCIÓN

Hoy se advierte una tendencia mundial para conseguir que los robots, sin importar el tipo, la plataforma o el sistema operativo que usen, estén en capacidad de comunicarse entre sí por medio de mensajes remotos intermediados por servidores [1], que estarían ejecutando servicios web [2] y que permitan desarrollar cualquier tarea propuesta de forma remota.

Uno de los principales aportes del manejo remoto de un robot es que, como existe un servidor o servidores remotos que intervienen en el proceso, es posible disminuir la carga computacional que el robot debe realizar [3], es decir, ya no sería el robot el encargado de los cálculos, análisis y ejecución de algoritmos, sino que parte de estas tareas serían asignadas a los servidores remotos como parte integrada del sistema robótico. Ello facilita el aumento de la capacidad de cómputo del sistema asociado con el robot para incrementar la

complejidad de los procesos y la capacidad de procesamiento por parte de los servidores asociados. Esto genera un nuevo tipo de robot, más económico, capaz de realizar innumerables tareas y para diferentes propósitos, haciéndolos más accesibles para el uso personal [4].

Este artículo se enfoca en mostrar que los servicios web semánticos [5] pueden ser aplicados para manipular remotamente robots construidos con la plataforma Arduino [6] para ejecutar jugadas básicas de fútbol [7]. Dichos servicios fueron descritos semánticamente con el lenguaje OWL-S [8] sobre una ontología de dominio [9] que describe un robot capaz de jugar fútbol. De este modo, se hace posible la invocación de estos servicios con el uso de la API de OWL-S [10].

Este documento está organizado de la siguiente manera: en la sección II se detalla la arquitectura propuesta para el sistema y se muestran las capas que la conforma. En la sección III se explica cómo se realizó la construcción de un robot futbolista con Arduino y sus principales características para ejecutar jugadas básicas de fútbol. En la sección IV se describe la implementación de los servicios web asociados con el sistema robótico, que facilitan la ejecución de jugadas de fútbol de manera remota. En la sección V se caracteriza la ontología de robots asociada con

el sistema robótico. Esta ontología describe un robot con sus características y componentes para jugar fútbol y se hace explícita la forma en la que se realizó la descripción semántica de los servicios. La sección VI presenta cómo se efectúa la invocación de los servicios a partir de su descripción semántica. Finalmente, en la sección VII, se presentan las conclusiones y trabajos futuros de este proyecto.

2. METODOLOGÍA PROPUESTA

Para soportar este trabajo se propone una metodología compuesta por los siguientes pasos: la implementación física del robot, la implementación de los servicios web que manejan remotamente el robot, la creación del marcado semántico de los servicios y la implementación del módulo para la invocación de los servicios. (Ver Fig. 1).



Fig. 1. Metodología propuesta

Como resultado de seguir esta metodología se obtuvo un sistema robótico manejado remotamente en el que el robot se comunica con el servicio web que, a su vez, tiene una descripción semántica que hace uso de una ontología del dominio en asociación con una ontología para la descripción de servicios y, finalmente, con el uso de las anteriores ontologías, se invoca cada servicio por medio de una aplicación orientada hacia el manejo de descripciones semánticas de servicios web.

3. CONSTRUCCIÓN FÍSICA DEL ROBOT FUTBOLISTA CON ARDUINO

El objetivo del robot es que sea capaz de realizar jugadas básicas de fútbol tales como atacar y defender con el uso de la plataforma de Arduino.

El robot fue construido con el ensamble de los siguientes componentes:

- **Placa Arduino mega ADK:** Contiene lo necesario para hacer funcionar el micro controlador; simplemente se conecta al computador con el cable USB o con un transformador o batería para empezar [11].
- **Bases en acrílico:** Soportes necesarios para poner los componentes electrónicos del robot. Se utilizaron 2 bases, una para soportar las ruedas y otra para las tarjetas controladoras del robot.
- **Ruedas omnidireccionales:** Cuando el eje gira, se comportan como una rueda normal. Sin embargo, su arquitectura permite que la rueda también se desplace hacia los lados.
- **Motoreductor con encoder:** Este motoreductor funciona con 12V y tiene una relación de reducción metálica de 50:1. Incluye un *encoder* de cuadratura que provee 64 pulsos por vuelta en el eje del motor y equivale a 3200 pulsos por vueltas en la salida de la caja de reducción.
- **Wireless Shield Proto:** Permite a la placa Arduino comunicarse de forma inalámbrica mediante un módulo inalámbrico.
- **Arduino Motor Shield:** Para manejar dos motores de corriente continua con la placa Arduino, el control de la velocidad y dirección de cada uno de forma independiente.
- **Brújula (compás) HMC6352:** Es un módulo para brújula completamente integrado que combina sensores magnético-resistivos de dos ejes con los circuitos análogos y digitales necesarios y los algoritmos para computación.
- **Dagu Mini Pan and Tilt Kit:** Este kit fue diseñado para hacer montajes de sensores y es adaptable a casi cualquier posición de los sensores que se requiera.
- **Sensor de color HDJD-S822:** Con un filtro de color rojo (R), verde (G) y azul (B) recubiertas por la matriz de fotodiodos, el sensor convierte la luz RGB para salidas analógicas de tensión.
- **Batería lipo KingMax:** La batería tiene dos celdas y la salida es de 7.4V 2200mAh de almacenamiento de carga.
- **Sensores infrarrojo:** Foto-transistores infra rojos de 940nm NPN 5mm oscuros, ampliados con electrónica (para que su capacidad de detección de la pelota sea mejor y mayor).
- **Polulo DC Motor Driver:** Compatible con ultrasonidos (hasta 20 kHz) de ancho de pulso (PWM) de la tensión de salida del motor que elimina la conmutación audible causada por el control de velocidad.

Con el uso de estos componentes se realizó el ensamblaje del robot y se llevaron a cabo los siguientes pasos:

- **Ensamble de la estructura de soporte de las ruedas:** Se ensambló la base de acrílico que sirve de soporte a las ruedas omnidireccionales y sus respectivos motores, como se detalla en la Fig. 2.



Fig. 2. Ruedas y motores sujetos a la base de acrílico

- **Ensamble de la base para el soporte de los dispositivos de control del robot:** En la segunda base de acrílico se ensambló la tarjeta controladora Arduino mega ADK y, luego, esta base se acopló a la base de las ruedas mediante un juego de tornillos dispuestos para este fin. (Ver Fig. 3).



Fig. 3. Arduino Mega ADK sujetado a la segunda base

- **Montaje en torre de las tarjetería de control Arduino:** Se montaron verticalmente las diferentes tarjetas controladoras asociadas con los periféricos del robot (*Wireless Shield Proto*, *Arduino Motor Shield*, Brújula (compás) HMC6352, *Dagu Mini Pan and Tilt Kit*, sensor de color HDJD-S822,

sensores infrarrojo), y se establecieron las conexiones respectivas con la tarjeta controladora mega ADK, previamente instalada. (Ver Fig. 4).

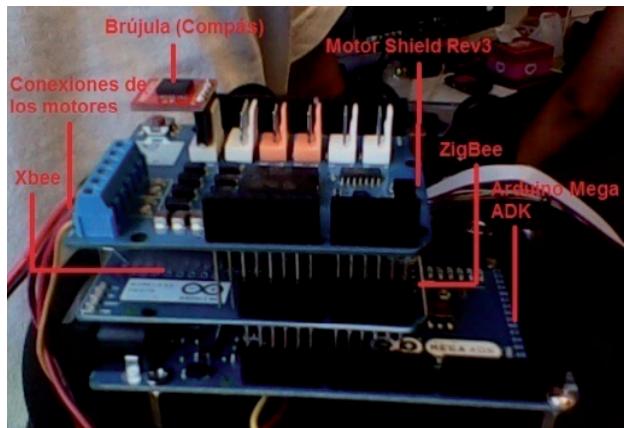


Fig. 4. Montaje en torre de la tarjetería restante

- **Ensamble del Dagu Mini Pan and Tilt Kit:** Se realizó el ensamblaje del manipulador que facilita la implementación del “brazo” para patear el balón de fútbol, que está montado en la parte delantera de la base que soporta las ruedas, como se muestra en la Fig. 5.



Fig. 5. Ensamble de la Dagu Mini en la parte delantera

Cuando esté listo el ensamblaje y la conexión de estos componentes, se obtiene el robot omnidireccional construido con la plataforma Arduino, capaz de ejecutar jugadas básicas de fútbol, como se advierte en la Fig. 6.

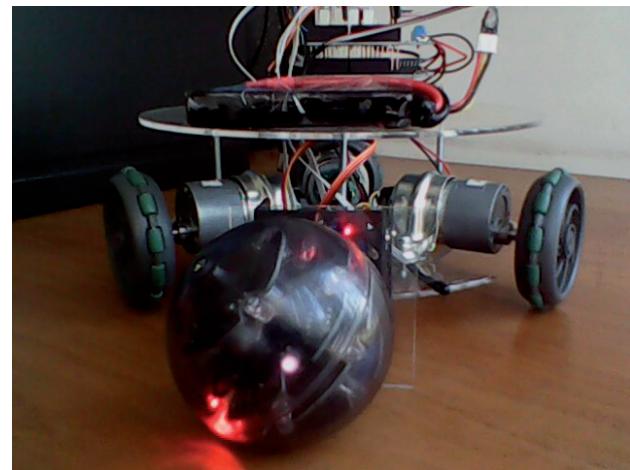


Fig. 6. Robot Arduino construido

4. CREACIÓN DE SERVICIOS WEB

Un servicio web es una aplicación ejecutable en un servidor y es accesible por medio de protocolos de comunicación, independientes de la plataforma o sistema operativo de quien realiza la comunicación ni del servidor en el que se encuentra alojado el servicio [15].

Si un servicio web es capaz de establecer una comunicación con un robot y enviarle a éste una acción para ejecutar, es posible conseguir que un robot sea controlado de manera remota. Los servicios web son un soporte básico de un área actual de investigación llamada robótica en la nube [16], en la que se desarrollan sistemas computacionales que permiten tener robots accesibles desde la web.

Para ver cómo apoyan los servicios web a la robótica y tener un primer acercamiento al concepto de la robótica en la nube, se decidió implementar tres servicios web para que hicieran tres acciones específicas de un robot jugador de fútbol: buscar la pelota, patear la pelota y pasar la pelota. Para estas acciones se implementaron los servicios de tal manera que reciben como parámetro la descripción de un robot y se basa en las habilidades, dispositivos y acciones del robot. El servicio determina si está en capacidad de llevar a cabo la acción solicitada. En caso de ser aprobada la solicitud, se realizan los cálculos pertinentes en el propio servidor y éste interactúa remotamente con el robot para llevar a cabo físicamente la acción. A continuación se describen los tres servicios relacionados:

- **El servicio BuscarPelota:** Este servicio determina si entre la descripción asociada con el robot se especifica que éste contiene una tarjeta de comunicación y si entre sus habilidades se encuentra la de buscar la pelota. En la figura 7 se muestra parte del código que realiza la esta tarea.

```

OWLKnowledgeBase kb = OWLFactory.createKB();
String[] reg = robot.split("<\"");
int inicio = reg[2].indexOf("\\"") + 1;
int fin = reg[2].lastIndexOf("\\"");
String uri = reg[2].substring(inicio, fin);
URI url = URI.create("http://localhost:8080/Trabajo 2 2 IW/buscarPelota.owl");
Service aService = kb.readService(url);
OWLIndividual individual = kb.getIndividual(URI.create(uri));
Object[] owl = individual.getProperties().values().toArray();
String salida = "";
for(int i = 0; i< owl.length; i++)
{
    salida += owl[i].toString().toLowerCase();
}
String respuesta = "El robot no es capaz de jugar futbol o buscar la pelota";
if(salida.contains("ruedas") && salida.contains("futbol") && salida.contains("x-bee") +
&& salida.contains("luz") && salida.contains("ultrasonido") && salida.contains("buscar")){
    respuesta = "El robot puede buscar la pelota";
}
return respuesta;
}

```

Fig. 7. Servicio web, BuscarPelota

- **El servicio PatearPelota:** Mediante este servicio se determina si en la descripción asociada con el robot se especifica que éste contiene una tarjeta de comunicación y si por lo menos una de sus habilidades es la de patear. En la Fig. 8 se muestran apartes del código que realiza la tarea.

```

OWLKnowledgeBase kb = OWLFactory.createKB();
String[] reg = robot.split("<\"");
int inicio = reg[2].indexOf("\\"") + 1;
int fin = reg[2].lastIndexOf("\\"");
String uri = reg[2].substring(inicio, fin);
URI url = URI.create("http://localhost:8080/Trabajo 2 2 IW/patearPelota.owl");
Service aService = kb.readService(url);
OWLIndividual individual = kb.getIndividual(URI.create(uri));
Object[] owl = individual.getProperties().values().toArray();
String salida = "";
for(int i = 0; i< owl.length; i++)
{
    salida += owl[i].toString().toLowerCase();
}
String respuesta = "El robot no es capaz de jugar futbol o patear la pelota";
if(salida.contains("ruedas") && salida.contains("futbol") && salida.contains("x-bee") +
&& salida.contains("luz") && salida.contains("ultrasonido") && salida.contains("ataque")){
    respuesta = "El robot puede patear la pelota";
}
return respuesta;
}

```

Fig. 8. Servicio web, PatearPelota

- **El servicio PasarPelota:** Este servicio define si el robot está en capacidad de pasar la pelota, por medio de un análisis de la descripción recibida, se verifica que entre sus habilidades se encuentre pasar la pelota y que, además, sea capaz de comunicarse por medio de una tarjeta. (Ver Fig. 9).

```

OWLKnowledgeBase kb = OWLFactory.createKB();
String[] reg = robot.split("<\"");
int inicio = reg[2].indexOf("\\"") + 1;
int fin = reg[2].lastIndexOf("\\"");
String uri = reg[2].substring(inicio, fin);
URI url = URI.create("http://localhost:8080/Trabajo 2 2 IW/pasarPelota.owl");
Service aService = kb.readService(url);
OWLIndividual individual = kb.getIndividual(URI.create(uri));
Object[] owl = individual.getProperties().values().toArray();
String salida = "";
for(int i = 0; i< owl.length; i++)
{
    salida += owl[i].toString().toLowerCase();
}
OWLIndividual individual2 = kb.getIndividual(URI.create("http://localhost:8080/Trabajo 2 2 IW/dominio.owl#Arduino"));
String respuesta = "El robot no es capaz de jugar futbol o pasar la pelota";
if(salida.contains("ruedas") && salida.contains("futbol") && salida.contains("x-bee") +
&& salida.contains("luz") && salida.contains("ultrasonido") && salida.contains("pase")){
    respuesta = individual2.toRDF(true, true);
}
return respuesta;
}

```

Fig. 9. Servicio web, PasarPelota

La implementación de estos servicios dejó entrever que es posible manejar remotamente un robot. Para ello, a cada servicio se le implementó inicialmente una comunicación mediante cable y luego, con una comunicación inalámbrica, se hizo uso de tarjetas XBee [18] entre el servidor web y el robot. Con ello, los procesos principales de cada jugada se realizan en el servidor y la programación básica asociada con el sistema operativo relacionada con movimientos y sensado, se efectúa en el robot.

5. DESCRIPCIÓN SEMÁNTICA DE LOS SERVICIOS

Con el fin de que las máquinas entendieran inequívocamente lo que hace cada servicio, se optó que para cada uno de los servicios implementados en la sección anterior se realizara su respectiva descripción semántica con el uso de ontologías. “Una ontología es una especificación explícita de una conceptualización”, según Thomas Gruber [12]. A continuación se describen los principales elementos que componen esta descripción semántica.

5.1 Creación de la ontología del dominio

Con el uso de lo que es una ontología, en esta sección se muestra que en el marco de este trabajo se realizó una representación del conocimiento asociado con el robot y el dominio de jugar fútbol. Como marco de referencia en la construcción de la ontología se asumió que un robot puede verse como una entidad a la que se le puede asociar un conjunto de habilidades para que ejecute tareas específicas como jugar fútbol al igual que se le asocia un conjunto de componentes mecánicos y electrónicos para que el robot lleve a cabo acciones de sensado, movimiento y planeación, entre otras. [13].

Para la construcción de la ontología se usó el editor de ontologías Protégé [14]. En la ontología se definió un robot como un objeto conformado por dispositivos, capaz de ejecutar diversas acciones, caracterizado por ciertas habilidades y de movilizarse según un sistema de locomoción. Los principales conceptos y las relaciones que conforman esta ontología son:

- **Los dispositivos:** Dentro de la ontología se definieron tres principales dispositivos (comunicación, energía y sensado). Cada uno de estos dispositivos contiene, a su vez, componentes más específicos para un robot, como se aprecia en la Fig.10.

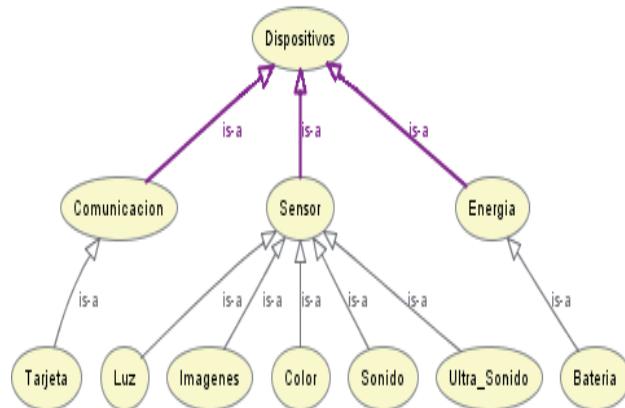


Fig. 10. Los dispositivos en la ontología

- **Las acciones:** Se definieron dos conjuntos básicos de acciones relacionadas con un robot capaz de jugar fútbol, como son las acciones de sensado y las acciones propias de las jugadas de fútbol que puede realizar el robot. Estas acciones se muestran en la Fig. 11.

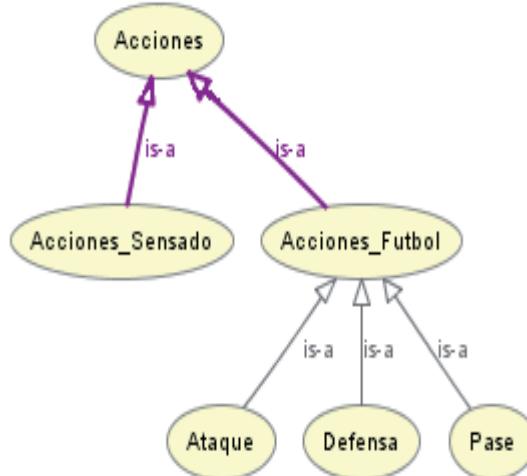


Fig. 11. Las acciones en la ontología

- **Las habilidades:** En la ontología asociada con el robot jugador de fútbol se definió el concepto de habilidades con el que se busca definir el concepto que reúne a aquellos elementos que ayuden al robot a llevar a cabo, con facilidad, cierta tarea. Para el robot futbolista se consideraron dos habilidades: la de juego y la de interacción con otros robots. En la Fig. 12 se muestran los conceptos y sus relaciones asociados con las habilidades del robot en la ontología.



Fig. 12. Las habilidades en la ontología

Finalmente, una visión general de la ontología se detalla en la Fig. 13.



Fig. 13. Ontología de un robot jugador de fútbol

5.2 Creación de la descripción semántica de los servicios

Para la descripción semántica de los servicios web se usó la ontología OWL-S [8]. Su implementación se hizo con el Plugin de OWL-S para Protégé [17]. Así, para cada servicio se definió su respectiva instancia OWL-S que hace uso de la ontología del do-

mino descrita en la sub-sección anterior con el fin de manejar los parámetros de entrada y salida de cada servicio. En la Fig. 14 se muestran apartes de la descripción semántica del servicio BuscarPelota en el que se observa la integración del OWL-S y la ontología del dominio asociada con las entradas de este servicio. (Ver el concepto *hasInput* resaltado en la Fig. 14). Tal como se aprecia en la figura, el servicio recibe como entrada el tipo de dato complejo Robot, que corresponde a la clase Robot_Móvil definida en la ontología del dominio.

```

<service:presents>
<profile:Profile rdf:ID="BuscarPelotaProfile">
<profile:hasInput>
<process:Input rdf:ID="Robots">
<process:parameterType rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI"
>http://localhost:8080/Trabajo_2_2_IW/dominio.owl#Robot_Móvil</process:parameterType>
</process:Input>
</profile:hasInput>
<service:presentedBy rdf:resource="#BuscarPelotaService"/>
<profile:textDescription rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
>Busca la pelota</profile:textDescription>
<profile:hasOutput rdf:resource="#Salida"/>
<profile:serviceName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
>Buscar Pelota</profile:serviceName>
</profile:Profile>
</service:presents>
  
```

Fig. 14. Descripción semántica de los servicios

6. INVOCACIÓN DE LOS SERVICIOS SEMÁNTICOS

Por tener una descripción semántica de los servicios, su invocación se puede realizar por medio de la API de OWL-S [10]. Esta API, permite crear una base de conocimiento soportada por las instancias de las ontologías de los servicios web y del dominio de fútbol que manejan. Esta API, basada en la descripción semántica de cada servicio web, permite obtener una instancia del servicio y del proceso asociado con éste, se definen las entradas (complejas o primitivas), las salidas requeridas y la ejecución del proceso asociado con cada servicio. En la Fig. 15 se muestran apartes del código en Java que, cuando se ejecuta mediante la API OWL-S, hace la invocación del servicio BuscarPelota.

```

public class BuscarPelotaInvocacion
{
    public static void main(String[] args) throws IOException, ExecutionException
    {
        ProcessExecutionEngine exec = OWLFactory.createExecutionEngine();
        OWLKnowledgeBase kb = OWLFactory.createKB();
        URI url = URI.create("http://localhost:8080/Trabajo 2 2 IW/buscarPelota.owl");
        Service aService = kb.readService(url);
        Process aProcess = aService.getProcess();
        ValueMap<Input, OWLValue> inputs = new ValueMap<Input, OWLValue>();
        inputs.setValue(aProcess.getInput(), kb.getIndividual(URI.create("http://localhost:8080/Trabajo 2 2 IW/dominio.owl#Arduino")));
        ValueMap<Output, OWLValue> outputs = exec.execute(aProcess, inputs, kb);
        OWLDataValue out = outputs.getDataValue(aProcess.getOutput());
        System.out.println(out.getValue());
    }
}

```

Fig. 15. Invocación del servicio con OWL-S API

8. CONCLUSIONES Y TRABAJO FUTURO

Con la implementación del prototipo del sistema de manipulación remota de robots por medio de servicios web semánticos, los usuarios, tanto humanos como las máquinas, pueden tener acceso al robot desde cualquier parte, siempre y cuando se tenga un punto de acceso a la web. Además, como se contempla una descripción semántica de los servicios que manipulan el robot, es posible realizar tareas de manera automática ya que las máquinas entienden la funcionalidad de los servicios que abren opciones interesantes como el descubrimiento automático de tareas que hace el robot al igual que componer automáticamente nuevas tareas que combinen los servicios asociados con las tareas que hace el robot.

Se encontró que existe la API OWL-S, que es una herramienta adecuada para invocar fácilmente servicios web marcados semánticamente. Esta API presenta un acoplamiento coherente con las descripciones de un dominio hechas en OWL y con las descripciones de servicios web con la ontología OWL-S. Su programación se hace en Java y se puede editar, invocar y crear servicios web desde cero.

El uso de servicios web para implementar en ellos los procesos complejos asociados con las tareas que debe realizar un sistema robótico y deja la implemen-

tación de pequeñas tareas básicas en el propio hardware del robot, permite que se disminuya el costo de los robots porque no se requeriría un hardware complejo en el propio robot.

Por último, la metodología propuesta en este trabajo es una muestra de cómo podría ser el proceso de construcción basado en una secuencia de capas que involucre la manipulación remota de un robot cualquiera.

Como trabajo futuro se propone hacer, de manera experimental, pruebas que permitan evaluar la complejidad y la eficiencia de éstos a clase de sistemas robóticos basados en servicios web semánticos orientados hacia el control remoto de los robots.

Otro trabajo futuro que se tiene proyectado es la implementación de un sistema capaz de componer tareas robóticas complejas a partir de tareas descritas mediante los servicios web semánticos. Para ello, se piensa utilizar algunas herramientas basadas en técnicas de planificación de inteligencia artificial adaptadas en la composición de servicios web.

Como última línea de trabajo futuro se plantea trabajar en mejorar nuevas técnicas de comunicación remota entre el servidor y el propio robot. Para ello, se considera estudiar nuevas tecnologías de comunicación inalámbrica como WI-FI y hacer uso de redes de telefonía celular para tener un mayor alcance entre el servidor y el robot.

AGRADECIMIENTOS

Este trabajo es apoyado por el proyecto “Programa de Fortalecimiento del Grupo de Investigación Sistemas Inteligentes Web – Sintelweb, convocatoria nacional 2010-2012, Modalidad 3”, con código 20201009532.

REFERENCIAS

- [1] Lafuente, A. Larrea, M, “Sistemas Distribuidos. Introducción”, Dpto. ATC UP/EHU, Jul. 10. 2009, Universidad del País Vasco. [en línea], Disponible en: <http://www.sc.ehu.es/acwlaalm/sdi/introduccioi-slides.pdf>. 2009.
- [2] Cavanaugh, E, “Web services: Benefits, challenges, and a unique, visual development solution”, white paper, Feb. 10. 2006, Sitio web de Altova. [en línea], Disponible en: <http://www.altova.com/whitepapers/webservice.pdf>. 2006.

- [3] Paul d. Hestand, a service oriented architecture for robotic platforms, university of massachusetts lowell, 2011
- [4] Willow Garage, "Personal Robot 2 (PR2)," [en línea], Disponible en: www.willowgarage.com.
- [5] Vaculin, R. Sycara, K, "Semantic Web Services Monitoring: AN OWL-S based Approach", 2007.
- [6] Margolis A, Arduino Cookbook, O'REILLY, 2011.
- [7] Matamoros, J M. López, J I, "Comportamiento adaptable de un equipo de fútbol de Robots Humanoides", 2011.
- [8] Martin D, Paolucci M, McIlraith S, Burstein M, McDenitt D, McGuinness D, Parsia B, Payne T, Sabou M, Solanki M, Srinivasan N & Sycara K. "Bringing Semantic to Web Services: The OWL-S Approach", 2004.
- [9] Boyce, S., & Pahl, C. Developing Domain Ontologies for Course Content. *Educational Technology & Society*, 10 (3), 275-288. 2007.
- [10] Sirin E & Parsia B. "The OWL-S Java API", 2004.
- [11] Arduino Mega ADK, Sitio web de Arduino [en línea], Disponible en: <http://arduino.cc/en/Main/ArduinoBoardADK>
- [12] Gruber, T, "What is an Ontology", 2007.
- [13] Schlenoff, C. Messina, E, "A robot Ontology for Urban Search and Rescue", 2006.
- [14] Noy. N F, McGuinness, D L, "Ontology Development 101: A Guide to Creating Your First Ontology", 2002.
- [15] Gunzer H, "Introduction to Web Services", 2002
- [16] Hu, G. Tay, W P. Wen Y, "Cloud Robotics: Architecture, Challenges and Applications", 2011.
- [17] Elenius D, Denker G, Martin D, Gilham F, Khouri J, Sadaati S & Senanayake R. "The OWL-S Editor A Development Tool for Semantic Web Services", 2005.
- [18] Mayalarp V, Limpaswadpaisarn N, Poombansao T & Kittipiyakul S. "Wireless Mesh Networking with XBee", 2010.
- [19] A. Elçi and B. Rahnama, "Human-Robot Interactive Communication Using Semantic Web Tech. in Design and Implementation of Collaboratively Working Robots", in Proc. ROMAN, pp.273-278. 2007.
- [20] Mortiz, T. Ulrich, K. Dejan, P & Micheale, B., "Web-enabled Robots that use the Web as an Information Resource", 2011.
- [21] Arkin, R., "Motor schema-based mobile robot navigation", In Proceedings of the IEEE International Conference on Robotics and Automation, pp.264–271. 2007.
- [22] Hogan, A. Harth, A. y Polleres, A., "Scalable authoritative OWL reasoning for the web", *Information Systems*, pp.49–90. 2008.
- [23] Gennari, J. Musen, M. Fergerson, R. Gross, W. Crubézy, M. Eriksson, H. Noy, N. y Tu, S., "The evolution of Protégé: an environment for knowledge-based systems development", *International Journal of Human-Computer Studies*, pp.89 – 123. 2003.