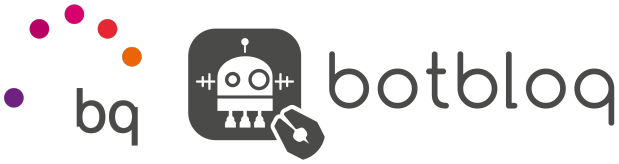

BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI)
EXPEDIENTE: IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

ACRÓNIMO DEL PROYECTO: BOTBLOQ

ENTREGABLE E.4.4 Definición alcance de la herramienta de software para la generación de estructuras robóticas.

RESUMEN DEL DOCUMENTO

Este documento define el alcance de la herramienta software a desarrollar para la creación de robots en el proyecto BOTBLOQ.

---

20 de enero de 2017

Índice

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

1. Introducción

Este documento forma parte de la documentación asociada al Paquete de Trabajo número 4 (PT4) del proyecto BOTBLOQ. El PT4 tiene como objetivo el diseño, la fabricación y el control de robots modulares. En los anteriores entregables se han realizado:

E.4.1. Informe detallado del estado del arte. Este documento contiene una clasificación de los robots modulares preexistentes y un análisis de los requisitos exigibles a una arquitectura robótica modular con el fin de definir las funcionalidades que debe reunir para que resulte satisfactoria según los objetivos del proyecto.

E.4.2. Diseño de la arquitectura modular. En este entregable se presentaron los módulos confeccionados para la implementación de estructuras robóticas, incluyendo características mecánicas, eléctricas, electrónicas y lógicas. En él también se describen los modos de generación permitidos (manual, predefinido y semi-automático), y se muestran ejemplos de robots configurables (predefinidos) con esta arquitectura.

E.4.3. Proceso metodológico para el diseño de un robot modular. Este entregable define las etapas a seguir para el diseño y programación de un robot en función de un conjunto de entradas.

En el presente documento se definirá el alcance y las posibilidades de la herramienta de software a realizar para ayuda al diseño de estructuras robóticas.

Para ello distinguiremos las siguientes capas en la aplicación:

Interfaz de usuario . Permite la interacción con el usuario. En dicha interfaz se deben proporcionar las distintas opciones necesarias para completar los pasos de la metodología de diseño indicada en el Entregable 4.3.

Consultas semánticas . Será necesario extraer información de la ontología creada en base al estándar del IEEE para Ontologías en Robótica y Automática [6] para poder generar la estructura abstracta del robot en base a los parámetros proporcionados en la interfaz de usuario.

Generación de la estructura física . Una vez se ha determinado la estructura abstracta, otra herramienta software debe generar automáticamente las piezas imprimibles en ficheros `.stl`.

Generación de controladores . Por último se ha de generar un modelo URDF y los controladores necesarios para realizar los comportamientos solicitados en la interfaz de usuario.

Aunque todas estas capas están relacionadas, pueden ser tratadas por separado. En las siguientes secciones se determinará el objetivo de cada una de estas partes del software a desarrollar.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

2. Interfaz

En esta sección se indicarán los requisitos exigibles a la interfaz de usuario del programa de diseño de robots para el proyecto BOTBLOQ.

La interfaz consistirá en una ventana en la que se deberán indicar los parámetros que se definen en las siguientes subsecciones: parámetros de entrada para la creación de la estructura abstracta y requisitos y parámetros físicos deseados en la estructura física. Estará escrita en lenguaje Python 3 para ser compatible con las consultas semánticas y con la programación de controladores.

2.1. Parámetros de entrada

Los parámetros de entrada son proporcionados por el usuario del sistema de diseño y sirven como punto de partida para la generación del robot. Dependiendo del modo de diseño, consistirán en:

- **Generación semi-automática:** El usuario indica las tareas/comportamientos que desea que el robot sea capaz de desempeñar (e.g.: correr). La interfaz podrá guiar al usuario entre el abanico de soluciones posibles.
- **Generación predefinida:** En este caso el usuario se limita a seleccionar un robot de entre una galería de robots predefinidos. El resto del proceso de diseño es bastante limitado, puesto que estos robots, (salvo por algunos parámetros opcionales), están completamente definidos de antemano (e.g. humanoide básico).

En el caso de generación manual, corre a cargo del usuario el diseño completo del robot, por lo que no se incluyen opciones para ello en la interfaz de usuario.

2.2. Requisitos/parámetros

Una vez se dispone de una estructura abstracta, se determinan los parámetros físicos necesarios para la definición de los componentes imprimibles mediante sistemas de prototipado rápido.

Requisitos. Especifican una restricción que condiciona la estructura mecánica/eléctrica del robot. Ejemplos de requisitos son:

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

- Velocidad a la que debe desplazarse un carro móvil.
- Máxima carga transportada por un manipulador industrial.
- Altura del escalón que debe ser capaz de subir un robot humanoide.

Parámetros. Especifican un valor necesario para definir completamente la estructura física del robot. Tienen valores por defecto que se aplican en caso de no ser especificados. Ejemplos de parámetros son:

- Longitud de las piernas de un humanoide.
- Número de módulos del cuerpo de un robot serpiente.
- Número de ejes de un carro móvil.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

3. Consultas semánticas

Una vez que el usuario ha facilitado los parámetros de entrada a la interfaz, el bloque de consultas semánticas debe explorar la ontología desarrollada para encontrar soluciones factibles para el conjunto de comportamientos solicitados.

La aplicación deberá ser capaz de realizar las siguientes operaciones, definidas en el proceso metodológico descrito en el Entregable 4.3:

1. Dado un conjunto de comportamientos por el usuario, se debe poder determinar, mediante consultas, las partes estructurales necesarias para ejecutar esos comportamientos.

2. Realizar matching entre una serie de partes estructurales y los robots capaces de realizar esas tareas según la ontología.

3. Guiar al usuario mediante una serie de cuestiones para desambiguar el resultado en el caso de que existen varias configuraciones capaces de desempeñar los comportamientos solicitados, de manera que se seleccione una sola configuración para el resto de los apartados de la metodología.

4. Generar una estructura abstracta, (instancia de la ontología) de la configuración seleccionada y proporcionar ese esquema al generador de la estructura física.

5. Seleccionar una estructura abstracta de entre las disponibles en la base de datos de configuraciones (instancias predefinidas de la ontología).

Las cuatro primeras operaciones son necesarias en el modo de diseño semiautomático, mientras que la quinta es la base del modo de diseño predefinido. En el modo de diseño manual la interfaz no generará una estructura abstracta, ya que el robot será completamente diseñado por el usuario.

Estas operaciones deben estar integradas dentro de la interfaz. La ontología de BOTBLOQ, ADROn (*Automatic Design of Robots Ontology*), ha sido generada siguiendo el estándar [6], el cual está escrito en lenguaje de representación de conocimiento SUO-KIF y basado en SUMO (Suggested Upper Merged Ontology, [7]). Por tanto, ADROn está escrita también en lenguaje SUO-KIF. La

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

librería pySUMO [2] permite realizar las consultas semánticas en el mismo lenguaje de programación la interfaz, Python 3, por lo que se utilizará para la integración de las mismas.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

4. Generación de la estructura física

Tras haber generado la estructura abstracta a partir de la ontología, la aplicación debe ser capaz de generar un modelo físico del robot. Este modelo físico incluye:

- Módulos pasivos necesarios para la construcción. Estos módulos se diseñarán expresamente para el robot diseñado en base a criterios dinámicos, de resistencia de materiales y de optimización de materia prima. El programa generará los ficheros de modelado sólido necesarios para la impresión de estas piezas (ficheros `.stl` de modelado 3D).
- Módulos activos necesarios para la construcción. Estos módulos se seleccionarán de entre la librería generada en el diseño de la arquitectura modular descrita en los ficheros. Cada módulo cuenta con sus propias instrucciones para su impresión y fabricación (por caras y unidos mediante colas de milano y tornillería).
- Sensores y actuadores necesarios para introducir en los módulos. Basados en el kit de Robótica de bq.
- Indicaciones de montaje mediante imágenes del robot finalizado.

El proceso para generar la estructura física necesita de una serie de parámetros adicionales.

4.1. Requisitos/parámetros

Una vez se dispone de una estructura abstracta, el usuario debe indicar una serie de requisitos y/o parámetros físicos adicionales para la definición de los componentes imprimibles mediante sistemas de prototipado rápido.

Requisitos. Especifican una restricción que condiciona la estructura mecánica/eléctrica del robot. Ejemplos de requisitos son:

- Velocidad a la que debe desplazarse un carro móvil.
- Máxima carga transportada por un manipulador industrial.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

- Altura del escalón que debe ser capaz de subir un robot humanoide.

Parámetros. Especifican un valor necesario para definir completamente la estructura física del robot. Tienen valores por defecto que se aplican en caso de no ser especificados. Ejemplos de parámetros son:

- Longitud de las piernas de un humanoide.
- Número de módulos del cuerpo de un robot serpiente.
- Número de ejes de un carro móvil.

En el Entregable 4.3 se describió el efecto de los requisitos dinámicos sobre la estructura física del robot (propiedades mecánicas) y se determinó una solución válida para el abanico de módulos disponibles en nuestra arquitectura. La aplicación debe incorporar dichos efectos para realizar un diseño robusto y efectivo.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

5. Generación de controladores

El programa debe ser capaz de permitir la programación de los comportamientos solicitados a los robots del modo más sencillo posible. Para ello la generación de la estructura física debe ir acompañada de un controlador que se encargue de generar las referencias de cada uno de los actuadores de forma que el robot realice un movimiento coordinado.

La creación de controladores para generar el movimiento en los robots es un problema arduo y con una casuística tremendamente extensa. Nuestra aplicación debe poder crear controladores parametrizables para cada uno de los robots de las estructuras abstractas obtenidas, de modo que los comportamientos puedan describirse a partir de las acciones básicas de movimiento/medida que pueda desarrollar cada robot, basándonos en el conjunto de sensores y actuadores de que disponga.

El desarrollo del controlador se realizará en ROS [3] mediante ya existentes en él, como puede ser el paquete *MoveIt!* [1], o mediante métodos propios (e.g. caminada de robots con patas de diferentes tamaños) y que generan unos controladores basados en las librerías de cinemática y dinámica KDL (*Kinematics and Dynamics Library* [4]) que incorpora ROS. Para ello será necesario haber generado un fichero URDF (*Unified Robot Description Format* [5]), que es el formato de descripción de robots preferido en ROS. Como alternativa a este diseño de control parametrizable, nuestro enfoque podría ser utilizado con configuraciones obtenidas por algoritmos evolutivos en un futuro.

Las capacidades de la aplicación para crear un controlador son las siguientes:

1. Generación del fichero URDF de descripción del robot a partir de la estructura física generada por la aplicación en el apartado anterior. Para esta generación se hace uso del lenguaje Xacro, que permite automatizar la creación de partes repetidas y parametrizar ciertos valores de los diseños.

2. Modelado del robot mediante la librería KDL a partir del fichero URDF.

3. Parametrización de los controladores en función de los requisitos/parámetros físicos indicados por el usuario.

4. Generación de bloques de código (funciones) para ejecución de comportamientos.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

5. Comunicación y programación del microcontrolador embebido en el robot (Intel Edison).

6. Integración con la interfaz de programación de bitbloq.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

Referencias

[1] MoveIt! `http://moveit.ros.org/`. Último acceso: 2015-12-23.

[2] pySUMO. `https://github.com/pySUMO/pysumo`. Último acceso: 2016-12-23.

[3] Robot Operating System. `http://www.ros.org/`. Último acceso: 2015-12-23.

[4] KDL ROS package. `http://wiki.ros.org/kdl`. Accessed: 2016-09-1.

[5] Unified Robot Description Format. `http://wiki.ros.org/urdf`. Accessed: 2016-09-1.

[6] IEEE Standard Ontologies for Robotics and Automation, 2015.

[7] A. Pease, I. Niles, and J. Li. The suggested upper merged ontology: A large ontology for the semantic web and its applications. In *In Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web*, 2002.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020