

# UNIVERSIDAD DE OVIEDO



Universidad de Oviedo



ASTURIAS  
CAMPUS DE EXCELENCIA  
INTERNACIONAL  
|AD FUTURUM|

## MÁSTER UNIVERSITARIO EN INGENIERÍA WEB

### TRABAJO FIN DE MÁSTER

#### TECNOLOGÍAS DE WEB SEMÁNTICA APLICADAS AL MODELADO DEL AGARRE DE OBJETOS



VºBº del Director del Proyecto

**DIRECTOR:** José Emilio Labra Gayo  
**AUTOR:** Francisco Javier Collado Valle

Oviedo, Julio 2014

## Resumen

La investigación presenta un modelo de datos para representar agarres de objetos que será una parte de un sistema formado por un módulo de visión artificial y un brazo con una mano robótica. El uso de manos robóticas hace que aparezca la necesidad de desarrollar un modelo de datos que permita no solo almacenar los agarres sino además poder relacionarlos con los objetos. Los objetos además deben ser reconocidos por el sistema de visión artificial, lo cual hace que sea necesario almacenar los datos que hacen que ésto sea posible. El agarre viene limitado por el tipo de mano robótica, i.e. número de dedos y grados de libertad, y a su vez el reconocimiento de objetos es dependiente del algoritmo utilizado. Dado que estos puntos son variables resulta necesario que el sistema sea adaptable y no estar ligado a un método concreto, para conseguir esta adaptabilidad y hacer que el sistema sea flexible y extensible se diseñó el modelo de datos utilizando tecnologías semánticas de modo que sea posible integrar nuevos algoritmos de reconocimiento de objetos o utilizar distintas taxonomías de agarres adecuados para la mano que se utilice en el sistema.

El modelo se basa en aislar las partes comunes a cualquier agarre, lo cual deja un núcleo formado por un pequeño conjunto de clases y propiedades, que las relacionan, dejando abierto el resto del modelo al sistema final. Para completar el modelo es necesario modelar los posibles agarres disponibles con la mano utilizada y modelar el almacenamiento de los datos necesarios para realizar el reconocimiento de objetos.

La investigación surge como parte de un proyecto en el que se utiliza una mano humanoide con 5 dedos capaz de ejecutar los agarres descritos por Cutkosky en su taxonomía de agarres. Por otra parte el sistema de reconocimiento de objetos se basó en varios algoritmos complementarios y todos ellos son modelados de diferente manera pero integrados en el modelo base.

El sistema final puede extender el modelo para adaptarlo a sus necesidades o modelar los agarres utilizando otra taxonomía de agarres diferente.

El modelo es capaz de integrar nuevos objetos de forma sencilla dado que los objetos se categorizan utilizando una jerarquía en función de sus características de modo que al integrar un nuevo objeto al sistema éste esté relacionado automáticamente con los agarres disponibles para la categoría a la que pertenece.

## Palabras Clave

Tecnologías semánticas, Ontologías, Objetos, Agarres, Visión Artificial, Mano Robótica

## **Abstract**

This research presents a data model for object grasping to be the knowledge base for a whole system composed by an computer vision module and a robotic arm with a robotic hand mounted on it. Robotic hands use pops up the need for developing a data model allowing not only to store the grasps but their relations with the objects. The objects must be recognised by the computer vision system, so the data to make this possible should be stored into the knowledge base too. Grasp type is constrained by the robotic hand type, i.e. number of fingers, degrees of freedom, and object recognition depends on the algorithm used by the system. Due to these variable things makes necessary an adaptable system to adapt itself to different grasp types and object recognition algorithms. To achieve this extensibility and adaptability the model is designed using semantic technologies, making possible the addition of new algorithms or the use of different grasp taxonomies, based on the hand used by the system.

The model is based on the isolation of common parts of the grasps, this isolation define a small set of classes and properties connecting them leaving the model open to the final system needs. To complete the model an adequate grasp taxonomy and the object recognition data information is required.

This research starts from a project using a five-fingered hand, which is able to execute grasps described by Cutcosky in his grasp taxonomy. The object recognition module is based on different, but complementary, algorithms with different data requirements but all of them integrated in the base model.

The final system can extend the model to adapt it to its own needs or describe available grasps using another grasp taxonomy.

This model is able to integrate new object in a simple way because objects are categorised using a hierarchy based on its characteristics, so the addition of a new object to the system makes an automatic relation with the grasps available for its category.

## **Keywords**

Semantic technologies, Ontologies, Objects, Grasp, Computer vision, Robotic hand

# Índice general

<b>1. Introducción</b>	<b>2</b>
1.1. Motivación del proyecto . . . . .	2
1.2. Finalidad del Proyecto . . . . .	3
<b>2. Fijación de Objetivos</b>	<b>4</b>
2.1. Posibles Ámbitos de Aplicación . . . . .	4
<b>3. Estado del arte</b>	<b>5</b>
3.1. Reconocimiento de Objetos . . . . .	5
3.1.1. Unión de esferas . . . . .	5
3.1.2. Cajas de volumen mínimo . . . . .	5
3.1.3. Geometría del objeto . . . . .	5
3.2. Identificación de agarres . . . . .	6
3.2.1. Agarres basados en la información visual . . . . .	6
3.2.2. Aproximación de zonas convexas . . . . .	6
3.2.3. Planificación de agarres . . . . .	6
3.2.4. Firmeza del agarre . . . . .	6
3.3. Base de datos de agarres . . . . .	6
3.3.1. Objetos en un entorno controlado . . . . .	6
<b>4. Descripción del Sistema</b>	<b>8</b>
4.1. Tecnologías utilizadas . . . . .	8
4.1.1. RDF . . . . .	8
4.1.2. OWL2 . . . . .	8
4.1.3. SPARQL . . . . .	8
4.1.4. Protégé . . . . .	9
4.1.5. Openlink Virtuoso . . . . .	9
4.1.6. ROS, Robot Operating System . . . . .	9
4.2. Descripción del modelo . . . . .	10
4.2.1. Modelo base . . . . .	10
4.2.2. Agarres . . . . .	10
4.2.3. Objetos . . . . .	12
4.2.4. Asignación de agarres . . . . .	12
<b>5. Metodología de Trabajo</b>	<b>15</b>

<b>6. Resultados Obtenidos</b>	<b>16</b>
6.1. Interpretación de los Resultados . . . . .	16
6.2. Discusión . . . . .	16
<b>7. Conclusiones y Trabajo Futuro</b>	<b>17</b>
7.1. Trabajo Futuro . . . . .	17
7.2. Difusión de los Resultados . . . . .	17
<b>Bibliografía</b>	<b>18</b>
<b>A. Artículos</b>	<b>20</b>
A.1. Expert Systems With Applications . . . . .	20

# Índice de figuras

4.1. Clases y propiedades del núcleo del modelo . . . . .	10
4.2. Taxonomía de agarres descrita por Cutkosky . . . . .	11
4.3. Modelado de la taxonomía de agarres de Cutkosky . . . . .	12
4.4. Taxonomía de objetos . . . . .	13

# Capítulo 1

## Introducción

La investigación se enmarca dentro del desarrollo del proyecto del séptimo programa marco CogLaboration. El proyecto consiste en llevar a cabo el intercambio fluido de objetos entre un humano y un brazo robótico, en el que se monta una mano robótica con cinco dedos similar a una mano humana. Para realizar el intercambio el sistema utiliza un módulo de visión artificial para el reconocimiento de los objetos de modo que el brazo robótico sea capaz de alcanzar y agarrar para posteriormente realizar el intercambio.

### 1.1. Motivación del proyecto

Para llevar a cabo el intercambio de objetos hay que tener en cuenta distintos factores como son que cada objeto requiere de un agarre diferente pero que a su vez un mismo objeto puede tener diferentes posibilidades de agarre. Por otra parte el entorno juega un papel importante a la hora de agarrar un objeto ya que el agarre más adecuado puede verse imposibilitado por la ubicación del objeto en el entorno, e.g. No sería lo mismo coger una llave posada en una mesa que si ésta cuelga de un clavo en la pared.

Determinar las posibilidades de agarre de un objeto no sólo se pueden basar en las propiedades del propio objeto (e.g. su forma) sino que la mano robótica que va a realizar el agarre también limitará los agarres que se pueden realizar, por lo que el modelo de datos debería ser capaz de poder adaptarse a los distintos tipos de manos robóticas de modo que una evolución en este componente no obligue a partir de cero de nuevo.

Por último el reconocimiento de objetos se basa en algoritmos que requieren unos datos diferentes (e.g. histogramas, modelos 3D, matrices de rotación,...) por lo que si el modelo no es capaz de adaptarse a distintos sistemas de reconocimiento de objetos se obliga a que cambiar el algoritmo de reconocimiento haga cambiar el modelo en el que basar el sistema completo.

## 1.2. Finalidad del Proyecto

Teniendo en cuenta lo anterior el proyecto trata de desarrollar un modelo de agarres de objetos que sea por una parte adaptable a distintas manos robóticas y por otra parte que permita su extensión para incorporar tanto nuevos objetos como diferentes algoritmos de reconocimiento de modo que estos cambios hagan que el modelo evolucione en lugar de producir un cambio que obligue a cambios importantes en el sistema.



# Capítulo 2

## Fijación de Objetivos

El proyecto aquí presentado encuentra su origen en el desarrollo del proyecto Coglaboration[1] pero su desarrollo trata de ir más allá haciendo que el modelo desarrollado para la base de conocimiento sea reutilizable y se pueda adaptar a proyectos con distinto enfoque, en cuanto a agarres o reconocimiento de objetos, pero con una finalidad semejante.

El objetivo principal es desarrollar el marco básico que permita modelar los agarres, los objetos y sus representaciones para que en función de las necesidades específicas las modificaciones y extensiones se integren en él con el menor esfuerzo posible.

Añadir nuevos objetos, y asignar sus posibles agarres, u obtener el agarre de objetos que no se encuentran en la base de conocimiento es un objetivo que permitirá hacer al sistema más autónomo y capaz de evolucionar con unas pequeñas modificaciones que no interfieran con el funcionamiento previo.

### 2.1. Posibles Ámbitos de Aplicación

El sistema se plantea como el asistente de una persona anciana a la hora de alcanzar objetos que necesite en su vida diaria (e.g. sartén, tazas, platos,...) o como el ayudante en ambientes profesionales para suministrar las herramientas necesarias para que un mecánico o un carpintero realicen sus tareas.

Dadas las diferencias existentes entre los posibles escenarios se hace aún más destacable la necesidad de que el modelado de agarres se pueda adaptar y extender de forma que el sistema se pueda integrar en los distintos ambientes.

# Capítulo 3

## Estado del arte

A continuación se muestran diversos métodos utilizados para el reconocimiento y agarre de objetos llevados a cabo en otros proyectos con diversos enfoques.

### 3.1. Reconocimiento de Objetos

#### 3.1.1. Unión de esferas

Para el reconocimiento de objetos Przybylski et al.[2] realizan una simplificación de los objetos aproximando su forma mediante la unión de esferas que permiten simplificar la evaluación del agarre más adecuado en función del objeto y la mano utilizada. Con el uso de esta técnica permite realizar agarres en objetos que no son conocidos a priori, pero al basarse en formas aproximadas hace imposible saber de qué objeto se trata y por tanto tener en cuenta restricciones de peso o rigidez del objeto que puedan dificultar su agarre.

#### 3.1.2. Cajas de volumen mínimo

Siguiendo la técnica de reconocimiento por aproximación a formas básicas Huebner et al.[3] realiza una descomposición utilizando cajas que aproximan la forma del objeto rodeando su superficie con el volumen mínimo necesario. Las cajas no son capaces de obtener una representación precisa del objeto, pero pueden ser utilizadas para planificar un agarre adecuado en objetos que no se conocen previamente. Al igual que en el caso anterior no existe relación entre objetos, por lo que la información aportada por el módulo de reconocimiento es la única de la que se dispone para realizar el agarre.

#### 3.1.3. Geometría del objeto

Basándose en la geometría del objeto y teniendo en cuenta su simetría nuevamente Przybylski et al.[4] describe un método para planificar los agarres de los objetos. En este caso se describen los objetos basándose en su volumen en lugar de utilizar otras representaciones (e.g. representaciones basadas en mallas 3D). Para llevar a cabo dicha representación de la superficie del objeto vuelve a utilizar esferas que tocan la superficie del objeto en uno o más puntos.

## 3.2. Identificación de agarres

### 3.2.1. Agarres basados en la información visual

Desde un punto de vista psicológico Lederman et al.[5] realiza una investigación sobre cómo la forma y la simetría de los objetos sugieren los agarres del objeto. La conclusión de dicho estudio es que las áreas convexas proporcionan agarres intuitivos a los humanos ya que sugieren al sujeto ejes de simetría a partir de dichas zonas convexas.

### 3.2.2. Aproximación de zonas convexas

Siguiendo la línea del estudio realizado por Lederman el método descrito por Harada et al.[6] utiliza modelos basados en la convexidad para evaluar los objetos y de esta forma calcular la posición de la mano y de los dedos para realizar el agarre. En este caso el sistema de reconocimiento de objetos no trata de reconocer el objeto como tal sino que lo simplifica basándose en la convexidad para encontrar los puntos de contacto que proporcionen el agarre más adecuado para el objeto.

### 3.2.3. Planificación de agarres

Para llevar a cabo la planificación de los agarres Jean-Philippe Saut et al.[7] introduce un método para realizar agarres con manos de varios dedos que se basa en calcular los agarres de independientemente del contexto, para ello sólo tiene en cuenta la mano que realizará el agarre y el objeto en lugar de evaluar un pequeño conjunto de agarres para un objeto dado. Este método permite calcular agarres en objetos conocidos o desconocidos, pero el tiempo requerido para realizar los cálculos es elevado y el propio autor recomienda el uso de un pequeño conjunto de objetos y agarres conocidos cuando se trabaja en entornos controlados.

### 3.2.4. Firmeza del agarre

A la hora de realizar un agarre la firmeza es una característica a tener en cuenta, por ello Kao et al.[8] presenta un modelo matemático para calibrar los agarres basándose en las habilidades de agarre de los humanos. En este caso la conocer la firmeza con la que realizar un agarre permitirá realizar agarres en objetos de distinta dureza de forma segura sin excederse de tal forma que el objeto quede firmemente agarrado pero no se llegue a dañar.

## 3.3. Base de datos de agarres

### 3.3.1. Objetos en un entorno controlado

Basándose en una base de datos Morales et al.[9] utiliza un conjunto conocido de objetos que el robot se encontrará en su entorno de trabajo para realizar los agarres. Junto con la base de datos que contiene las descripciones de los objetos y sus agarres el

sistema cuenta con un módulo de análisis de agarres, que provee el mejor agarre basado en simulaciones, y un módulo de visión estereoscópica encargado de identificar y localizar el objeto. La limitación de este sistema viene dada porque no es capaz de agarrar ningún objeto que no haya sido previamente calculado mediante las simulaciones.

# Capítulo 4

## Descripción del Sistema

### 4.1. Tecnologías utilizadas

#### 4.1.1. RDF

Modelo para el intercambio de datos en la web estandarizado por el W3C[10]. La estructura del RDF está basada en tripletas, cada tripleta está compuesta de sujeto, predicado y objeto. La estructura del RDF enlaza cada sujeto con sus propiedades.

Dado que RDF está orientado al intercambio de datos en la web la identificación de sujetos, propiedades y objetos se realiza mediante URIs para hacerlos unívocos. La estructura del RDF representa un grafo dirigido donde las aristas son los predicados y los nodos son los recursos, i.e. sujetos y objetos.

Una de las mayores ventajas del RDF sobre el XML es la posibilidad de unir datos de diferentes esquemas y la posibilidad de modificar el esquema sin la necesidad de modificar los clientes que consumen los datos. Ésta posibilidad hace que RDF se ajuste a lo que se busca en el proyecto para poder extender y adaptar el modelo de agarres a las necesidades de cada sistema.

#### 4.1.2. OWL2

El modelo está basado en OWL2[11], una familia de lenguajes para la representación de conocimiento y desarrollo de ontologías desarrollado y estandarizado por el W3C. Las ontologías son vocabularios formales de términos que cubren generalmente un dominio específico y que son compartidos por un grupo de usuarios. Las ontologías OWL pueden usarse junto con información descrita en RDF y ser a su vez exportadas como documentos RDF.

#### 4.1.3. SPARQL

Un *endpoint* SPARQL es un servicio que cumple con el protocolo SPARQL[12] y permite a sus usuarios, ya sean humanos o máquinas, consultar una base de conocimiento

utilizando el lenguaje SPARQL[13] y recibir los resultados en formatos procesables por ordenador. Normalmente, tanto la consulta como la representación de los resultados son realizados utilizando módulos software que muestran esos resultados en un formato legible por humanos, i.e. páginas web, PDF, hojas de cálculo.

La consulta es realizada usando el lenguaje SPARQL, que es un lenguaje estandarizado para consultar grafos RDF. Esta característica hace que la modificación del modelo sea más fácil que en otros sistemas de almacenamiento como las bases de datos relacionales tradicionales, donde el modelo de datos es más rígido.

### 4.1.4. Protégé

Para el desarrollo del modelo se ha utilizado Protégé, un editor de ontologías desarrollado por la Universidad de Stanford en colaboración con la Universidad de Manchester. Además de permitir la edición de ontologías Protégé también es un sistema de gestión y adquisición de conocimiento. Desarrollado en Java con interfaz gráfica Swing proporciona un framework para el desarrollo de extensiones para ampliar su funcionalidad.

De acuerdo con su sitio web[14] Protégé tiene más de 200.000 usuarios y es considerado el editor de ontologías más popular.

### 4.1.5. Openlink Virtuoso

Openlink Virtuoso es la versión libre de Virtuoso Universal Server[15], el producto comercial desarrollado por Openlink Software. Virtuoso combina diferentes funcionalidades como Sistema Gestor de Bases de Datos Relacional, Servidor de Ficheros y Aplicaciones Web, Almacén de Tripletas RDF/XML y *Endpoint* SPARQL.

En el desarrollo del proyecto se utiliza Virtuoso para el almacenamiento del modelo desarrollado y de las tripletas que definen los datos de agarres y objetos. Por otra parte también se utiliza el *endpoint* SPARQL para realizar las consultas que devuelvan los datos.

### 4.1.6. ROS, Robot Operating System

Aunque no es directamente aplicable a lo presentado en este proyecto, puesto que no se muestra la integración final del modelo en un sistema real, se ha utilizado ROS[16] para permitir la comunicación de los módulos de visión artificial y manejo del brazo y manos robóticas con la base de conocimiento.

Para poder consultar el *endpoint* SPARQL desde los distintos módulos se desarrollaron unos servicios escritos en Python que reciben los datos de la consulta para posteriormente devolver los resultados en el formato requerido por dichos módulos. Dichos servicios se desarrollaron a modo de prueba de concepto para comprobar la posibilidad de integrar la base de conocimiento basado en tecnologías semánticas con un sistema robótico real.

### 4.2. Descripción del modelo

A continuación se describe el modelo desarrollado diferenciando tres partes. En primer lugar se describe el modelo base para a continuación describir el modelo de agarres y por último la jerarquía de objetos utilizados.

#### 4.2.1. Modelo base

La base del modelo desarrollado, como se puede ver en la Figura 4.1, se centra en el agarre de objetos por lo que la clase central del modelo es *Grasp*, representando el agarre que se realiza sobre el objeto. Por una parte el agarre está relacionado con un tipo de agarre, representado en la clase *GraspType* que será la clase padre de todos los tipos de agarres posibles, por otra parte la clase *Grasp* mantiene relación mediante la propiedad *hasGrasp* con los objetos que, como se verá posteriormente, estarán representados como subclases de la clase *Object*. Como se vio en el Estado del arte (Capítulo 3) para realizar un agarre hay que tener en cuenta la posición de la mano respecto del objeto, lo cual se representa en el modelo mediante la clase *WristPose* y que será descrito mediante subclases y subpropiedades según el brazo sobre el que esté montada la mano y que no será objeto de estudio en este proyecto.

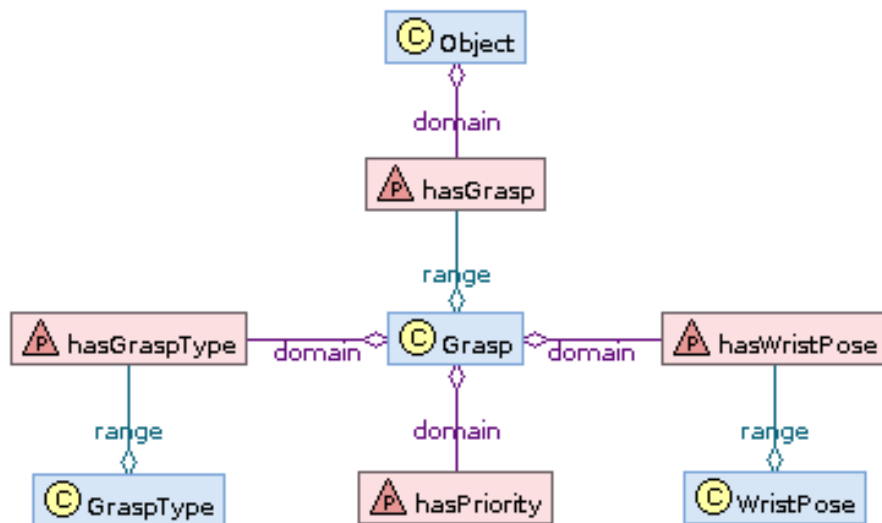


Figura 4.1: Clases y propiedades del núcleo del modelo

#### 4.2.2. Agarres

Como se comentó anteriormente el tipo y número de agarres que un sistema robótico puede realizar viene determinado principalmente por la mano robótica que se utilice. En la actualidad se pueden encontrar en el mercado manos robóticas de cinco dedos similares, tanto en peso como en tamaño, a la mano humana (e.g. Shadow Dexterous Hand[17] Prensilia IH2 Azzurra[18]). Dado que en el proyecto CogLaboration se utiliza una mano

de estas características se optó por realizar un modelo de tipos de agarres basado en la taxonomía de agarres descrita por Cutkosky[19].

Cutkosky en su taxonomía de agarres realiza una clasificación (Figura 4.2) distinguiendo principalmente agarres de precisión de agarre, aquellos que requieren principalmente destreza y sensibilidad, de los agarres fuertes, que requieren seguridad y estabilidad. Las sucesivas divisiones van diferenciando el número de dedos implicados o la disposición y número de dedos implicados en el agarre.

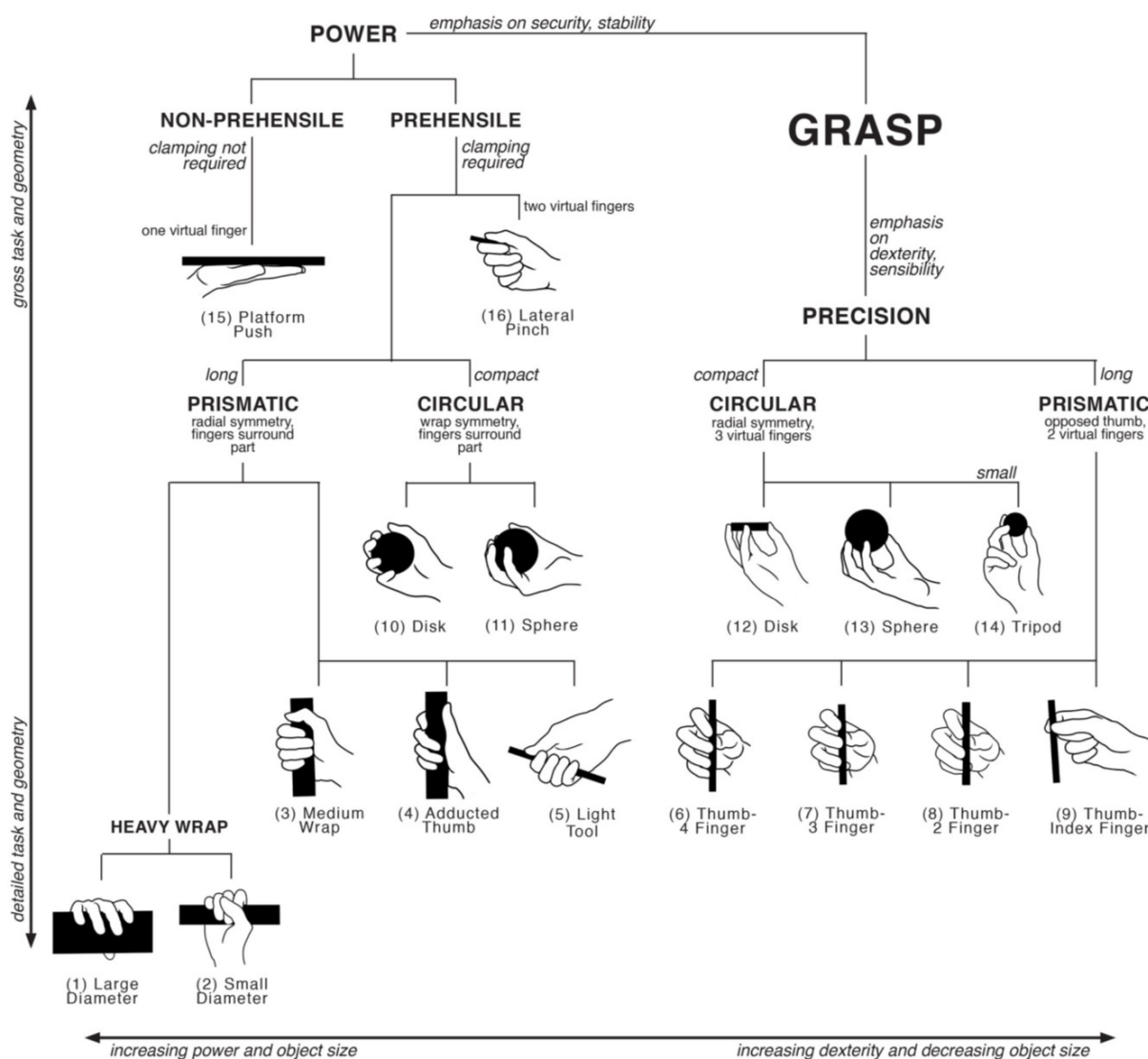


Figura 4.2: Taxonomía de agarres descrita por Cutkosky

Para modelar dicha taxonomía e integrarla en el modelo base se crearon subclases de la clase *GrapsType* con las categorías de agarres descritas por Cutkosky para finalmente representar los dieciséis agarres como individuos de la ontología. (Figura 4.3).



## 4. Descripción del Sistema

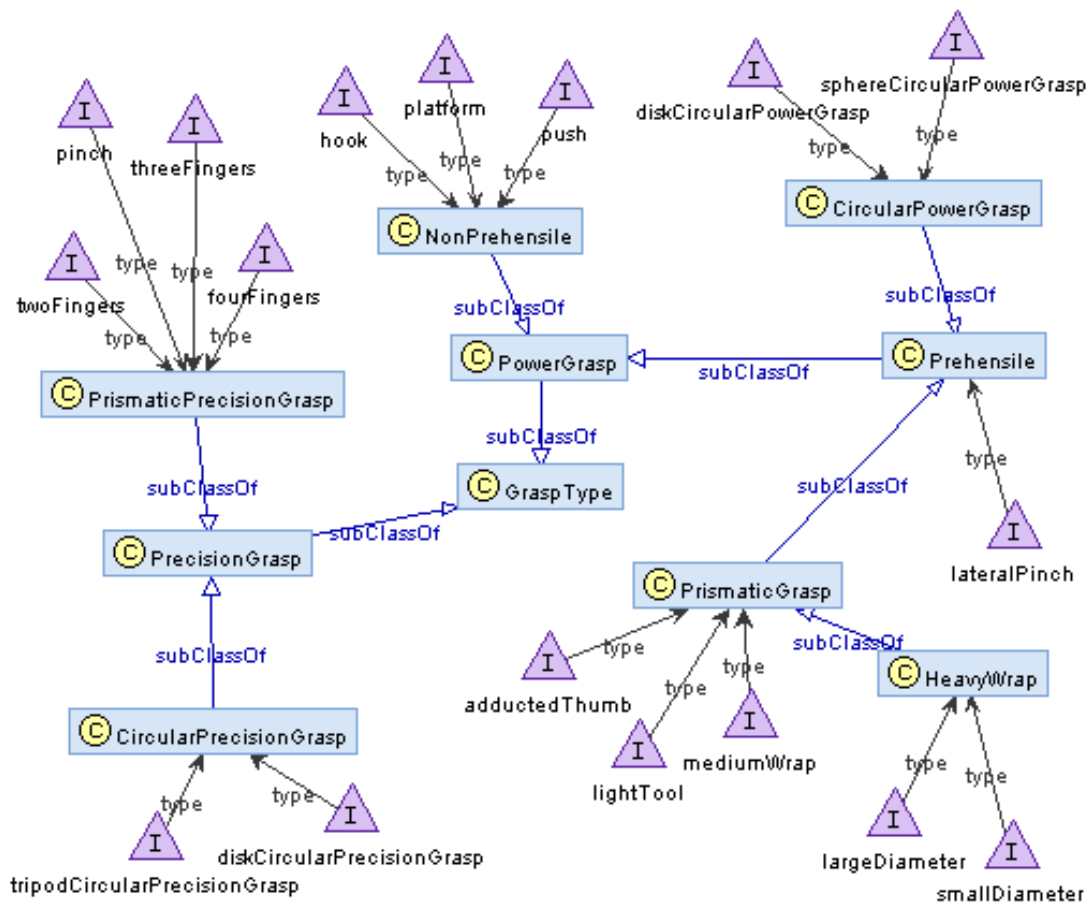


Figura 4.3: Modelado de la taxonomía de agarres de Cutkosky

### 4.2.3. Objetos

De forma semejante a como se modelan los agarres concretos se realiza un modelado de los objetos creando una jerarquía a partir de la clase *Object*. Para el desarrollo del proyecto se optó por crear una jerarquía basada en la zona de agarre de los mismos (Figura 4.4). En cuanto a la zona de agarre se diferencian tres tipos, objetos con empuñadura, objetos con asa y objetos, herramientas en este caso, con mango. La elección de esta característica no implica que sea su única forma de agarre, pero sí que dichas zonas de agarre se presentan como una posibilidad evidente. Dado que existen multitud de objetos que pueden no presentar una zona de agarre tan definida también se optó por realizar una clasificación de formas básicas, en la que algunos objetos pueden encajar directamente, i.e. un mando a distancia podría clasificarse, de cara a su agarre, como un prisma.

### 4.2.4. Asignación de agarres

Una vez creado el modelo básico, la jerarquía de agarres y la jerarquía de objetos los agarres (*Grasp*) se clasifican creando una jerarquía paralela a la de objetos de forma que si existe una clase *Cup* exista una clase *CupGrasp* relacionada con ella que tenga un

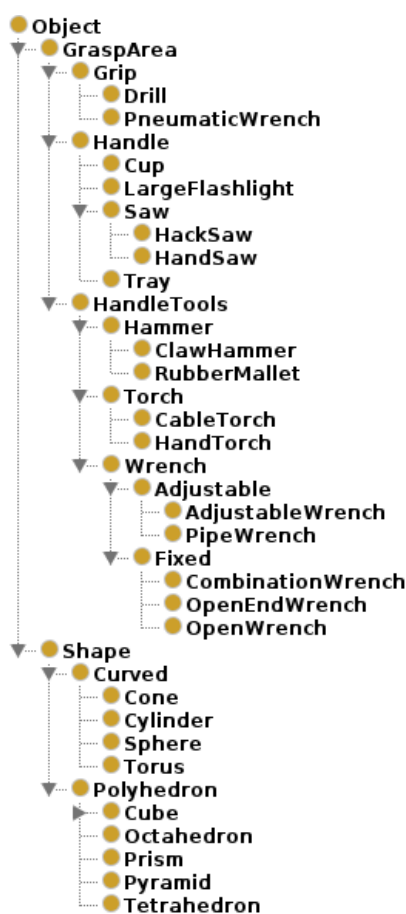


Figura 4.4: Taxonomía de objetos

*GrasType*, una *WristPose* y una prioridad.

Esta forma de asignar los agarres a los objetos permite que un nuevo objeto incluido en la base de conocimiento bajo una categoría tenga automáticamente asociados los agarres de dicha categoría, y en caso de ser necesario podrían inferirse sus agarres en base a los agarres de las categorías padre, es decir, agarres menos específicos.

Dentro de cada categoría de la jerarquía de objetos se crean instancias, i.e. individuos, que representan un objeto concreto del mundo real y que será reconocido por el sistema de visión artificial. Estos individuos, como se explica anteriormente, no requieren de la asignación de agarres concretos, ya que su pertenencia a una categoría de la jerarquía es la que hace que sus agarres sean definidos, llevándolo a un caso concreto y simplificado podríamos entender que si las tazas tienen un solo agarre, i.e. agarrar por el asa, todos los individuos de la categoría taza serán agarrados por su asa. Los individuos de la clase taza podrían tener diferencias entre sí, interesantes para el reconocimiento, pero en todo caso todos los individuos de dicha categoría tendrían en común el ser objetos con un asa lateral situada de forma vertical.

## 4. Descripción del Sistema

---

### Consulta SPARQL

En base a la categorización de los objetos concretos el sistema podrá recuperar los agarres de un individuo concreto realizando una consulta SPARQL a la base de conocimiento. Una vez reconocido el objeto, bien sea un objeto concreto existente en la base de conocimiento o una forma genérica, el módulo encargado de realizar la consulta utilizará su identificador para obtener aquellos agarres asignados a su categoría.

A continuación se muestra la consulta SPARQL (Listing 4.1) que se utiliza para obtener el tipo de agarre, la prioridad y la posición de la muñeca para el objeto (i.e. [objectname]) que se indique.

```
PREFIX :<http://www.javicollado.com/instances.owl#>
PREFIX kb:<http://www.javicollado.com/knowledgebase.owl#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?graspTypeID ?priority ?wristPose WHERE {
    :objectname a ?class .
    ?subGrasp rdfs:subPropertyOf kb:hasGrasp .
    ?subGrasp rdfs:domain ?class .
    ?subGrasp rdfs:range ?graspClass .
    ?grasp a ?graspClass .
    ?grasp kb:hasGraspType ?graspType .
    ?grasp kb:hasPriority ?priority .
    ?grasp kb:hasWristPose ?wristPose .
    ?graspType kb:hasGraspTypeNumber ?graspTypeID .
}
```

Listing 4.1: Consulta para obtener el identificador del tipo de agarre, la prioridad y la posición de la muñeca para un objeto dado

Nótese que en la consulta se devuelve *?wristPose* como resultado para la posición de la muñeca, aunque en un sistema real esa posición de la muñeca estará almacenada de la forma que el sistema requiera, e.g. matriz de rotación de la muñeca frente al objeto, y por tanto la consulta debería obtener ese dato a partir del *?wristPose* que aquí se deja como resultado.

# Capítulo 5

## Metodología de Trabajo

Para el desarrollo del modelo se parte de una revisión de la literatura existente sobre sistemas de reconocimiento de objetos cuya finalidad es el agarre. Una vez obtenidos diversos sistemas de referencia se traslada dicha información al proyecto en desarrollo tratando de extraer de ellos aquellas partes que son comunes y que permiten describir los agarres.

Como se comentó con anterioridad, el número y tipo de agarres viene condicionado por el tipo de mano robótica empleada, por lo que una vez determinada la mano a utilizar en el proyecto (i.e. IH2 Azzurra[18]) se procede al modelado de los distintos tipos de agarres posibles. Dejando este punto integrado en el modelo pero agrupado bajo una clase común (*Grasp Type*) que permite el intercambio o extensión para adaptarlo a distintas manos o sistemas.

Una vez desarrollada la parte de los tipos de agarre se procede a crear una jerarquía de objetos. Para realizar dicha jerarquía se escoge un conjunto variado de objetos que puedan existir en los escenarios planteados (i.e. hogar y taller) y se agrupan en base a distintas características. En un primer momento se realiza una agrupación por tamaño, para posteriormente realizar la agrupación por peso. Con el avance del módulo de reconocimiento de objetos se termina optando por una agrupación basada en características visuales únicamente y, dado que se busca conseguir su agarre, se categorizan en función de su principal modo de agarre (i.e. empuñadura, asa, mango,...).

Tras el desarrollo del modelo se comienza a introducir datos sobre diferentes objetos, extraídos del módulo de visión artificial, y a asignarles los agarres correspondientes, basados en los datos proporcionados por los socios del proyecto.

Dado que el modelo debería integrarse en el sistema robótico se procede al desarrollo de los servicios que proporcionen la interfaz entre ROS y el *endpoint* SPARQL, de modo que se pueda comprobar el correcto funcionamiento del modelo en un entorno cada vez más próximo a lo que será su fin último, que es proporcionar una base de conocimiento al sistema completo.

# Capítulo 6

## Resultados Obtenidos

La ontología presentada en este proyecto trata de crear un marco de trabajo base para el desarrollo de modelos de agarre adaptables y extensibles. El modelo básico consta del conjunto mínimo de clases y propiedades para describir agarres de objetos de una forma básica, de forma que la implantación en un sistema real requiera del modelado de las características concretas del sistema de destino.

### 6.1. Interpretación de los Resultados

El modelo aquí descrito sirvió de base para el desarrollo de la base de conocimiento del proyecto CogLaboration[20]. Dicha base de conocimiento partió del núcleo de clases y propiedades aquí descrito, así como de las jerarquías de agarres y de objetos, para posteriormente adaptar y extender el modelo según las necesidades que fueron surgiendo durante el proyecto.

Las modificaciones y ampliaciones realizadas en la ontología para llegar a utilizarla en un sistema real, en ningún caso afectaron al conjunto básico que se estableció como conjunto común a cualquier sistema, por lo que de este modo se puede entender que se han alcanzado los resultados buscados desde un primer momento para que el modelo no quedase ligado a un sistema concreto.

### 6.2. Discusión

Dado que durante la revisión de la literatura existente apareció el concepto de simetría de los objetos y su relación con los agarres, se trató de integrar dicha característica en el modelo base, aunque finalmente no fue posible aislar de forma clara el concepto para que fuese reutilizable debido a la dependencia encontrada de la simetría de los objetos y el método de reconocimiento objetos.

# Capítulo 7

## Conclusiones y Trabajo Futuro

Pese a nacer en el contexto de un proyecto más amplio, desde un principio se trató de aislar el modelo del resto del sistema, de modo que su reutilización fuera posible. Dado que se inicia su desarrollo basándose en datos de otros sistemas que sólo comparten la finalidad de realizar agarres con el proyecto de destino, se considera un éxito haber logrado que el modelo aquí descrito haya sido utilizado satisfactoriamente como base de conocimiento en CogLaboration, demostrando así su adaptabilidad.

Dado que CogLaboration tenía como objetivo no sólo el agarre de los objetos, sino su intercambio con humanos, el hecho de haber podido ampliar el modelo base para incluir la información relativa al intercambio, i.e. entrega y recepción, hace pensar que la extensión del modelo es posible, más allá del ámbito de los agarres, hacia otros ámbitos donde los agarres sean una parte del proceso.

Dotar de semántica al modelo permitió que incluir nuevos datos en la base de conocimiento haya sido una tarea menos laboriosa y que el reconocimiento de objetos previamente desconocidos no suponga la exclusión del objeto o una limitación a la hora de elegir un agarre adecuado.

### 7.1. Trabajo Futuro

Puesto que el modelo ha sido ampliado para poder integrarlo en un sistema como el presentado por CogLaboration, el futuro del modelo pasaría por revisar esas extensiones para tratar de integrarlas en el modelo de agarres y de este modo hacer que el modelo no sólo describa agarres sino que también sirva para modelar intercambios manteniendo la flexibilidad y la capacidad de extensión que por el momento se ha conseguido en una primera fase.

### 7.2. Difusión de los Resultados

Tras la realización de la investigación se ha enviado un artículo a la publicación *Expert Systems With Applications* describiendo el modelo desarrollado. (véase apéndice A.1)

# Bibliografía

- [1] Coglaboration. successful real world human-robot collaboration: From the cognition of human-human collaboration to the cognition of fluent human-robot collaboration (November 2011).  
URL [http://cordis.europa.eu/projects/rcn/101415\\_en.html](http://cordis.europa.eu/projects/rcn/101415_en.html)
- [2] M. Przybylski, T. Asfour, R. Dillmann, Unions of balls for shape approximation in robot grasping, in: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, 2010, pp. 1592–1599.
- [3] K. Huebner, S. Ruthotto, D. Kragic, Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping, in: 2008 IEEE International Conference on Robotics and Automation (ICRA 2008), 2008.
- [4] M. Przybylski, T. Asfour, R. Dillmann, Planning grasps for robotic hands using a novel object representation based on the medial axis transform, in: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, IEEE, 2011, pp. 1781–1788.
- [5] S. J. Lederman, A. M. Wing, Perceptual judgement, grasp point selection and object symmetry, *Experimental Brain Research* 152 (2) (2003) 156–165.
- [6] K. Harada, K. Kaneko, F. Kanehiro, Fast grasp planning for hand-arm systems based on convex model, in: 2008 IEEE International Conference in Robotics and Automation, 2008.
- [7] J.-P. Saut, D. Sidobre, Efficient models for grasp planning with a multi-fingered hand, *Robotics and Autonomous Systems* 60 (3) (2012) 347–357.
- [8] I. Kao, M. R. Cutkosky, R. S. Johansson, Robotic stiffness control and calibration as applied to human grasping tasks, *Robotics and Automation, IEEE Transactions on* 13 (4) (1997) 557–566.
- [9] A. Morales, T. Asfour, P. Azad, S. Knoop, R. Dillmann, Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands, in: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, IEEE, 2006, pp. 5663–5668.
- [10] W3C, Rdf 1.1 concepts and abstract syntax (2 2014).  
URL <http://www.w3.org/TR/rdf11-concepts/>

- [11] W3C, Owl 2 web ontology language document overview (12 2012).  
URL <http://www.w3.org/TR/owl2-overview/>
- [12] W3C, Sparql 1.1 protocol (3 2013).  
URL <http://www.w3.org/TR/2013/REC-sparql11-protocol-20130321/>
- [13] W3C, Sparql 1.1 overview (3 2013).  
URL <http://www.w3.org/TR/sparql11-overview/>
- [14] Protégé (July 2014).  
URL <http://protege.stanford.edu/>
- [15] Openlink virtuoso universal server (July 2014).  
URL <http://virtuoso.openlinksw.com/>
- [16] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, Ros: an open-source robot operating system, in: ICRA workshop on open source software, Vol. 3, 2009, p. 5.
- [17] A. Kochan, Shadow delivers first hand, *Industrial robot: an international journal* 32 (1) (2005) 15–16.
- [18] M. Controzzi, C. Cipriani, M. C. Carrozza, Mechatronic design of a transradial cybernetic hand, in: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, IEEE, 2008, pp. 576–581.
- [19] M. R. Cutkosky, On grasp choice, grasp models, and the design of hands for manufacturing tasks, *Robotics and Automation, IEEE Transactions on* 5 (3) (1989) 269–279.
- [20] D. Cabañeros, D3.40 knowledgebase for object features storage and management (6 2014).  
URL [http://coglaboration.eu/sites/default/files/CogLaboration\\_{\\_}D3.40.pdf](http://coglaboration.eu/sites/default/files/CogLaboration_{_}D3.40.pdf)



# Apéndice A

## Artículos

### A.1. Expert Systems With Applications

El artículo *Adaptable And Extensible Grasping Framework Using Semantic Technologies* ha sido enviado a la publicación *Expert Systems With Applications*.

# Adaptable And Extensible Grasping Framework Using Semantic Technologies

Javier Collado<sup>a</sup>, José Emilio Labra<sup>b</sup>

*Universidad de Oviedo*

<sup>a</sup>*uo173921@uniovi.es*

<sup>b</sup>*labra@uniovi.es*

---

## Abstract

In this paper we present a grasp model framework. The use of a robotic hand to grasp objects pops up the need for defining how the hand would grasp the object, but also requires the object recognition to be defined and modelled. The grasp is constraint by the robotic hand, i.e. number of finger and DOF. The object recognition could be very different with regard to the algorithms used. This framework is built using semantic technologies to provide flexibility to integrate with different kind of systems, giving the chance to adapt the model to different hand abilities or object recognition algorithms.

Basing the model on different taxonomies the core is reduced to a small set of classes and properties. The final system should use the core to implement a specific knowledge base creating subclasses and subproperties or extending the model to fulfil its needs.

*Keywords:* Semantic technologies, Ontologies, Objects, Grasp, Computer vision, Robotic hand

---

## 1. Introduction

There are a lot of improvements in the technological world to make more accessible and more useful products. Elder people are a big example of people needing aid to complete some daily tasks.

Robotic technologies are a huge mix of several different areas joint to achieve one goal. Robots are widely used in industrial environment on automated tasks, but the idea of a robot helping people at home is far from being a reality. At home the scenario is completely different because the

environment is not controlled at all and the required tasks will not be fully automated, so the robot has to adapt its knowledge.

These handicaps should be gotten over by the robots to enter into people's homes and they should mimic human behaviour to integrate themselves into people's life. One of these handicaps is object grasping and exchange, different objects require different grasp but the optimal grasp may be impossible due to other objects near the target one. If these tasks were completed in a natural way the human and robot collaboration would be better, but as mentioned before, the robot have to deal with a changing environment and different kind of tasks (e.g. take a mug from the dishwasher or pick up a knife from the drawer).

There are multiple situations where a robot could be helpful but adaptation and extensibility are the key to be useful in the long run.

We could imagine an old woman living alone and a robot helping her to cook. The help could be the robot giving her the pan and the ingredients or reaching those kitchen tools stored too high for her.

Our goal is to develop an ontology capable of describe objects and their grasp to be the foundation for a robotic system with the ability to grasp and exchange objects with a human, but a closed system will not be the answer, we need to provide the possibility to grasp the unknown object and still grasp it in a natural way.

## 2. Related work

Due to the nature of the grasping task there are different things to take into account. The robot should detect the target using a computer vision module and here we can find some projects generating an image joining basic forms. Przybylski et al.[10] take this way and represent the objects as a union of balls to simplify the evaluation to obtain the most adequate grasp regarding the object and the robot's hand. This technique let integrate unknown objects to the system and grasp them using their external shape, but, as an unknown object, the system can't take into account the weight or the stiffness of the object.

This shape approximation was also used by Hueber et al.[5] but instead of balls they do a decomposition using different-size bounding boxes using the minimum volume required to represent the objects, i.e. the boxes surround the object to represent its surface. Those boxes, as they say, are not able to approximate the objects in a precise manner but can be used to determine

an adequate grasp on unknown objects.

Following the use of balls to represent the objects Przybylski et al.[11] developed another grasp planning algorithm based on object's geometry and taking into account the symmetry properties of the object. This object representation describes the object's geometry in terms of volume as opposed to other representations, e.g. mesh representation, that describe the object's surface. In this case the representation uses spheres touching the boundary of the object at one or more points.

With regard to the grasp planning Jean-Philippe Saut et al.[13] introduced a method for a multi-fingered hands to compute a context-independent set of grasps, i.e. they only take into account the object and the robot hand to compute the grasps instead of a small set of optimal grasps for a given object. In this case the algorithm can compute the grasp for any object, either known or unknown object, but computation time is not negligible and they suggest the use of a small set of known objects and grasps when working in controlled and known environments.

As humans we rely in visual information to grasp an object and Lederman et al.[8] investigate about how object shape and symmetry suggest the grasp for a given object. Their study also concludes that objects with convex areas provide an intuitive grasp because people can figure out a symmetry axis from this convex areas.

In the same line of Lederman's investigation Harada et al.[4] described a method to evaluate objects as a set of convex models that approximate the actual object. Using these convex models they can calculate the finger and hand position to grasp the object. In this case the computer vision algorithm instead of recognise the actual object tries to simplify the object generating the convex models in order to find the contact points to effect a proper grasp.

Regarding to the stiffness of the grasp of an object Kao et al.[6] presents a mathematical model to calibrate grasping based on human grasping abilities. Once the robot can identify and grasp objects the stiffness of the grasp is a basic parameter to achieve good results when grasping different kind of objects.

Another approach is the use of a database, Morales et al.[9] decide to fill it with the description of all the objects present in the robot workspace. With the database there are a grasp analysis system and a stereo vision system, the former providing the best grasp using simulations and the latter is responsible of localize the object. This framework is capable of plan grasps after recognise the object in the workspace.

### 3. System overview

Working in a controlled environment provides the security of a closed object set which can be precomputed to have the best grasp for each situation, but to extend the set the system needs to calculate the best grasp for the new object. This problem arise because each new object is something completely different, without connection with previous knowledge. Not only this knowledge connection is important, the object representation could be based on different methods and this methods could be complementaries. Due to this need for flexibility the system would be flexible enough to accept different or new object recognition methods

Our knowledge base is based on semantic technologies, in particular the knowledge base model is described using OWL ontologies[14], with the intention of achieve the, previously commented, extensibility and adaptation to the unknown objects and new situations required by the open environment. First of all the knowledge will be presented as an ontology to express the current relations amongst grasps, objects, grasp type and object representations.

Due to the different robotic hands in the market the available grasps will be determined by the robotic hand in use. Nowadays we can find five-fingered hands (e.g. Shadow Dexterous Hand[7], Prensilia IH2 Azzurra[2]), similar to human hand. These kind of hands can reproduce human movements and human grasps, the different grasps available in this hands were identified and categorised into a taxonomy developed by Cutkosky[3], this classification differentiate, as shown in Figure 1, stable and secure grasps, i.e. power grasps, from those grasps requiring dexterity and sensibility, i.e. precision grasps. Cutkosky's taxonomy was chosen to represent the available grasps in our system and represented as a class hierarchy in the ontology. The different grasp types described by Cutkosky are represented as individuals which type is a subclass of *GraspType* representing the different categories expressed in the original taxonomy(Figure 2).

Once we had defined the class *GraspType* the next step was to define all the actors involved in a grasp, and with that base integrate and develop the complete model. The framework core (Figure 3) has four classes and four properties to define a grasp. A grasp type is required to define a grasp, but the grasp must be related with an object, the wrist pose to achieve the grasp

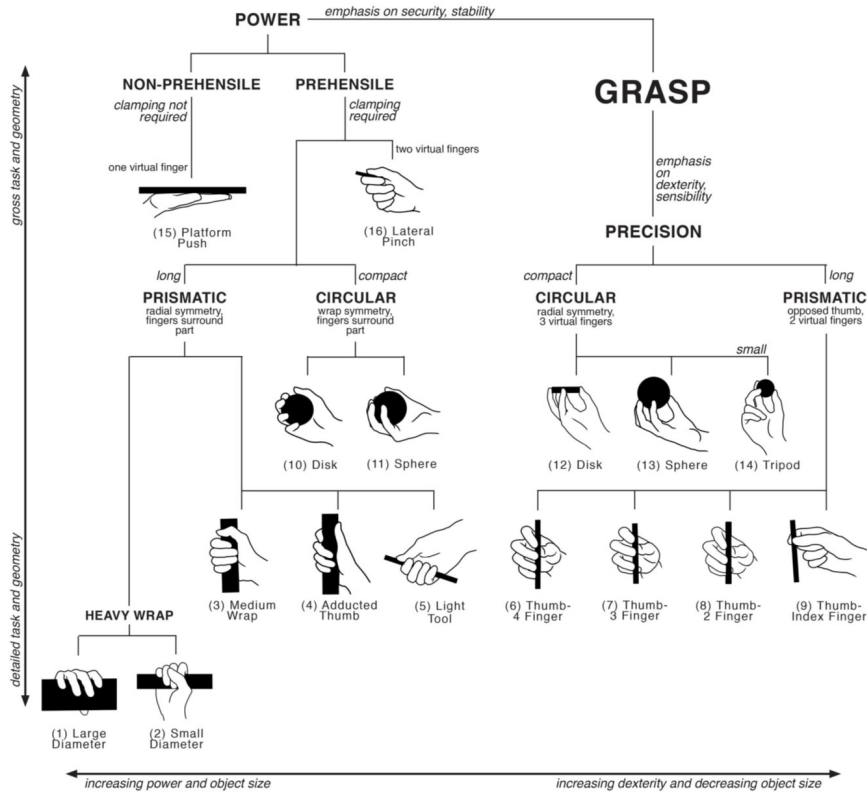


Figure 1: Cutkosky's Grasp Taxonomy.

and the priority for the given object because one object could be grasp in more than one way but some of them could be better than the others.

On one hand, the *Object* class is the root of an object taxonomy that should be completed with object categories, i.e. an object could present a handle (e.g. hammer, drill) or could be a basic shape (e.g. cylinder, cube, sphere). The more specific categories are defined the better semantic meaning will have the model. Individual objects will be integrated into the hierarchy based on their properties and features. On the other hand the *Grasp* class is the root of a grasp taxonomy that will be developed in parallel with the object taxonomy, i.e. if there is a cup then a cup grasp should be defined.

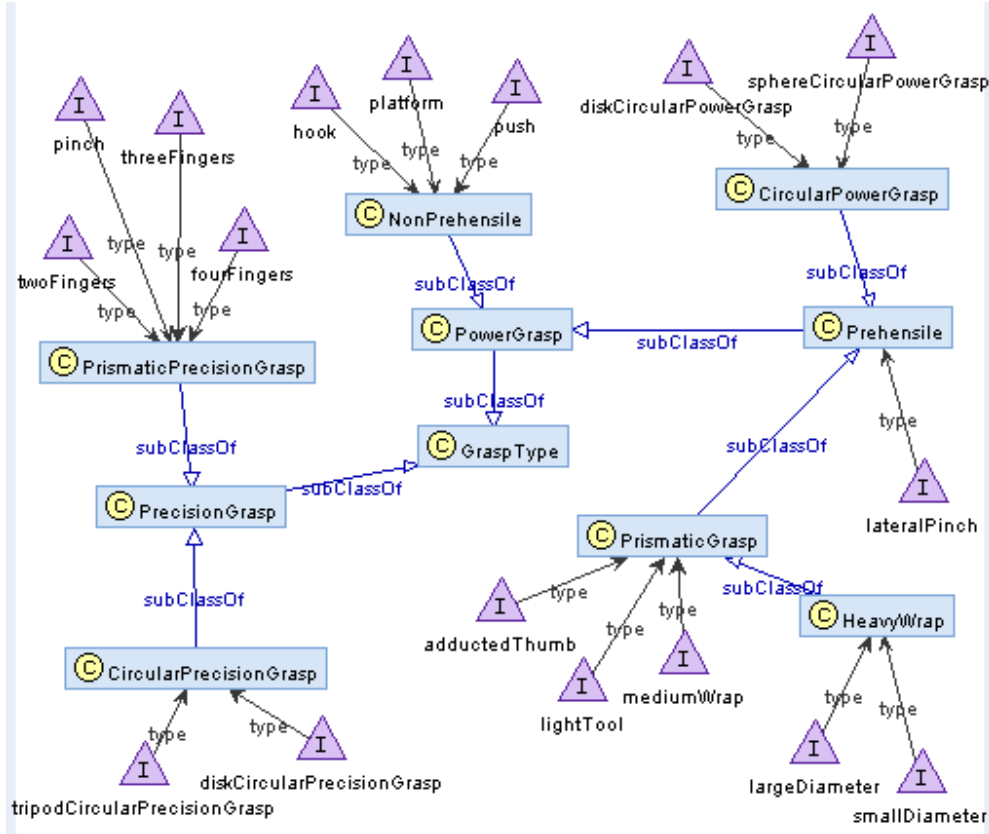


Figure 2: Class Hierarchy Representing Cutkosky's Grasp Taxonomy.

### 3.1. Matching grasps and objects

As mentioned before, one of our goals is to simplify the definition of the grasp for unknown objects, with the model described above and an object taxonomy we can infer the grasp for an unknown object simply recognising that object as one of the defined categories. Object individuals will populate the different taxonomy categories. Each individual will adopt the grasps assigned to its category, e.g. when we insert a new individual *green mug* under the category *Cup*, with its own visual model to allow its recognition, the individual will have all the grasps related with cup category. This situation let fill the knowledge base with new objects with ease, and also let the system to grasp an unknown object using its form, e.g. if the object is not recognised as a specific object but the object match with a basic shape, such

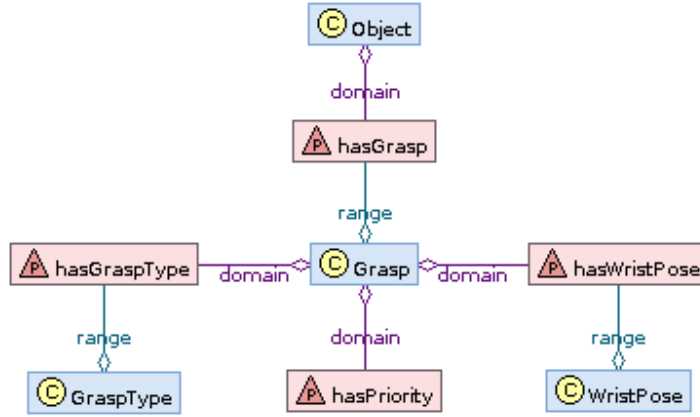


Figure 3: Ontological grasping framework base.

as a cylinder or a cube, the knowledge base could provide an adequate grasp for this shape.

These assignments are made because as soon as a new object is categorised as a *Cup* the system relates the new individual with all the grasps defined as *CupGrasp*, even if the object is recognised as a basic shape, due to the impossibility of a more precise recognition, the system will return the different grasp possibilities for that shape, e.g. if the system recognises a *Cube* then the *CubeGrasps* will be returned when required. To complete this matching the system will use an SPARQL endpoint[15][16] to store the model and the data using RDF[17].

The use of RDF, OWL and SPARQL let extend the system to adapt the model to the definition of new objects and shapes or even change the grasp taxonomy used. The model extension could provide new parameters to define grasps without breaking the previous knowledge, so the knowledge base could store legacy data aside with new data allowing the existence of different systems over the same knowledge base reusing all the model but the object recognition system representation.

#### 4. Future work

This model is intended to be integrated into a robotic system to manage the knowledge that will be used by the object recognition system and the



robotic hand.

To test this model the next step is the integration with ROS[12], an open-source Robot Operating System. This integration could be achieved developing some ROS services to provide the information required by other robot modules, i.e. hand controller and object recognition module. These services would query the knowledge base using the data collected by the recognition module and return the most accurate results to the controller.

With regard to the extension, adaptation and integration with ROS the model described in this paper was the foundation for the CogLaboration knowledge base[1] development. Using this framework as a core then the project has implemented different extensions and adapting the model to achieve their goals.

Another improvement over this first grasping model would be introduce the hand over information to complete the object exchange between the robotic hand and a human. Not only hand over information could be useful, some objects may have some manoeuvrability constraints, e.g. a cup must be hold with the top looking up to avoid spill its content.

Since CogLaboration goes further with the model extension than the creation of its own object taxonomy and integrates a first approach to the hand over modelling, it could be interesting the integration of these model to extend the framework core to allow hand over an object manoeuvrability based on this development.

## 5. Conclusions

Regarding to the object-grasping knowledge base, and due to the improvements made both in robotic hands and object recognition, the model flexibility and adaptation is a desirable feature which allow the model to improve and share information amongst different systems without been outdated with those research advances.

On the other side, semantic relations between objects and grasps could provide information about specific unknown objects with similar shape. Even when object recognition could become less dependant of specific objects and recognise them as a family of objects without the worry of be unable to find a suitable grasp.

## References

- [1] Cabañeros, D., 2014. D3.40 knowledgebase for object features storage and management. URL: [http://coglaboration.eu/sites/default/files/CogLaboration\\_D3.40.pdf](http://coglaboration.eu/sites/default/files/CogLaboration_D3.40.pdf).
- [2] Controzzi, M., Cipriani, C., Carrozza, M.C., 2008. Mechatronic design of a transradial cybernetic hand, in: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, IEEE. pp. 576–581.
- [3] Cutkosky, M.R., 1989. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *Robotics and Automation, IEEE Transactions on* 5, 269–279.
- [4] Harada, K., Kaneko, K., Kanehiro, F., 2008. Fast grasp planning for hand-arm systems based on convex model, in: *2008 IEEE International Conference in Robotics and Automation*.
- [5] Huebner, K., Ruthotto, S., Kragic, D., 2008. Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping, in: *2008 IEEE International Conference on Robotics and Automation (ICRA 2008)*.
- [6] Kao, I., Cutkosky, M.R., Johansson, R.S., 1997. Robotic stiffness control and calibration as applied to human grasping tasks. *Robotics and Automation, IEEE Transactions on* 13, 557–566.
- [7] Kochan, A., 2005. Shadow delivers first hand. *Industrial robot: an international journal* 32, 15–16.
- [8] Lederman, S.J., Wing, A.M., 2003. Perceptual judgement, grasp point selection and object symmetry. *Experimental Brain Research* 152, 156–165.
- [9] Morales, A., Asfour, T., Azad, P., Knoop, S., Dillmann, R., 2006. Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands, in: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, IEEE. pp. 5663–5668.

- [10] Przybylski, M., Asfour, T., Dillmann, R., 2010. Unions of balls for shape approximation in robot grasping, in: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE. pp. 1592–1599.
- [11] Przybylski, M., Asfour, T., Dillmann, R., 2011. Planning grasps for robotic hands using a novel object representation based on the medial axis transform, in: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, IEEE. pp. 1781–1788.
- [12] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y., 2009. Ros: an open-source robot operating system, in: ICRA workshop on open source software, p. 5.
- [13] Saut, J.P., Sidobre, D., 2012. Efficient models for grasp planning with a multi-fingered hand. *Robotics and Autonomous Systems* 60, 347–357.
- [14] W3C, 2012. Owl 2 web ontology language document overview. URL: <http://www.w3.org/TR/owl2-overview/>.
- [15] W3C, 2013a. Sparql 1.1 overview. URL: <http://www.w3.org/TR/sparql11-overview/>.
- [16] W3C, 2013b. Sparql 1.1 protocol. URL: <http://www.w3.org/TR/2013/REC-sparql11-protocol-20130321/>.
- [17] W3C, 2014. Rdf 1.1 concepts and abstract syntax. URL: <http://www.w3.org/TR/rdf11-concepts/>.