# Modelo de Aplicaciones Sensibles al Contexto (MASCO), caso de estudio: Robot móvil recolector de objetos

Evelina C. Velázquez[1], María del Pilar Gálvez D.[1], Ariel N. Guzman Palomino[1]

[1] Facultad de Ingeniería, Universidad Nacional de Jujuy, Provincia de Jujuy, Argentina.
caro_eve@hotmail.com, mdpgalvezdiaz@fi.unju.com.ar, nel_gp@hotmail.com.

**Resumen.** El presente trabajo tiene como finalidad la adaptación del Modelo de Aplicaciones Sensibles al Contexto (MASCO) del Grupo de Investigación GRISECO de la Facultad de Ingeniería de la Universidad Nacional de Jujuy, al caso de estudio: Robot móvil recolector de objetos. Se aprovecharon las características del modelo para desarrollar un prototipo robótico cuya función es el proceso de recolección de un objeto que es detectado en la trayectoria de movimiento de un robot dentro de una habitación.

**Keywords:** Contexto, MASCO, Robótica, Robot.

## 1 Introducción

La Robótica ha sufrido un gran progreso desde sus inicios en muy pocas décadas, esto es debido a la capacidad de los robots de cumplir tareas críticas y también tediosas para el hombre, en una cada vez más extensa variedad de áreas. Esto es posible a través de sensores y actuadores que conectan a un robot con su entorno. El presente trabajo consiste en la adaptación del Modelo MASCO al desarrollo de un robot de movimiento que posee un brazo robótico con el cual realiza el agarre de objetos, cuya ubicación es determinada por un sensor sonar incluido en el robot, los objetos son colocados en un compartimento en el mismo robot, para que éste cumpla su función de recolección.

El apartado 1 corresponde a la introducción, el 2 consiste en una introducción a los conceptos básicos de Robótica, el apartado 3 define las capas del Modelo MASCO, el cual es adaptado en el apartado 4 al caso de estudio, donde además se indican consideraciones generales de los movimientos que realiza el robot y el cálculo de los ángulos que posiciona al brazo para realizar el agarre del objeto. El apartado 5 contiene las conclusiones y el 6 las referencias bibliográficas.

## 2 Robótica

La Robótica ha cobrado gran importancia en la actualidad, debido a sus múltiples usos en diferentes áreas, desde usos espaciales, medicina, industria y también usos sociales, entre otros. Esta se define "como el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano o que requieren del uso de inteligencia" [1]

En el ámbito social, hay innovaciones con respecto a las aplicaciones de la Robótica, tanto de asistencia a personas con capacidades disminuidas y también en tareas diarias.

Todas las innovaciones se lograron a partir de la percepción de variables a través de dispositivos como sensores, y de las funciones que pueden llevar a cabo los robots. Para clarificar, se define robot como: *"*Manipulador multifuncional y reprogramable, diseñado para mover materiales, piezas, herramientas o dispositivos especiales, mediante movimientos programados y variables que permiten llevar a cabo diversas tareas" según la definición de Robot del Instituto Norteamericano de Robótica. Dado que esta definición se aplica sobre todo a robots industriales, sería más apropiado definirlo como: "un dispositivo electromecánico multipropósito capaz de reaccionar a estímulos captados por sensores, los que son interpretados y que componen el entorno donde se encuentra, mediante el uso de actuadores" [3].

Utilizando sensores y actuadores, los robots pueden ejecutar funciones con información del ambiente que los rodea, tomando variables de interés del contexto, estas son procesadas y tras una decisión, se ejecutan funciones a través de actuadores que permiten al robot interactuar con un ambiente. Se definen los sensores como dispositivos capaces de captar y traducir información en valores que son compresibles u operables, y los actuadores dispositivos de salida que permiten al robot realizar acciones. Así los sensores otorgan al robot la capacidad de tomar valores de interés del ambiente que lo rodea, y los actuadores le permiten realizar tareas en consecuencia para cumplir un objetivo.

## 3 Modelo de Aplicaciones Sensibles al Contexto (MASCO)

El Modelo para aplicaciones Sensibles al Contexto (MASCO) es una solución para desarrollar sistemas sensibles al contexto (Fig. 1), realizado por el grupo de investigación GRISECO (Grupo de Ingeniería de Aplicaciones Sensibles al Contexto) de la Facultad de Ingeniería de la Universidad Nacional de Jujuy. [4]

MASCO consta de 5 niveles:

- Aplication Layer: consta de los objetos del dominio de la aplicación.
- Context Layer: en esta capa se encuentran los objetos que procesan la información de las otras capas.
- Service Layer: los objetos de esta capa brindan servicios.
- Sensing Concerns Layer: posee objetos que se encargan de la interpretación los datos de la capa inferior adyacente.
- Hardware Abstractions Layer: contiene a los objetos que representan los sensores y actuadores.

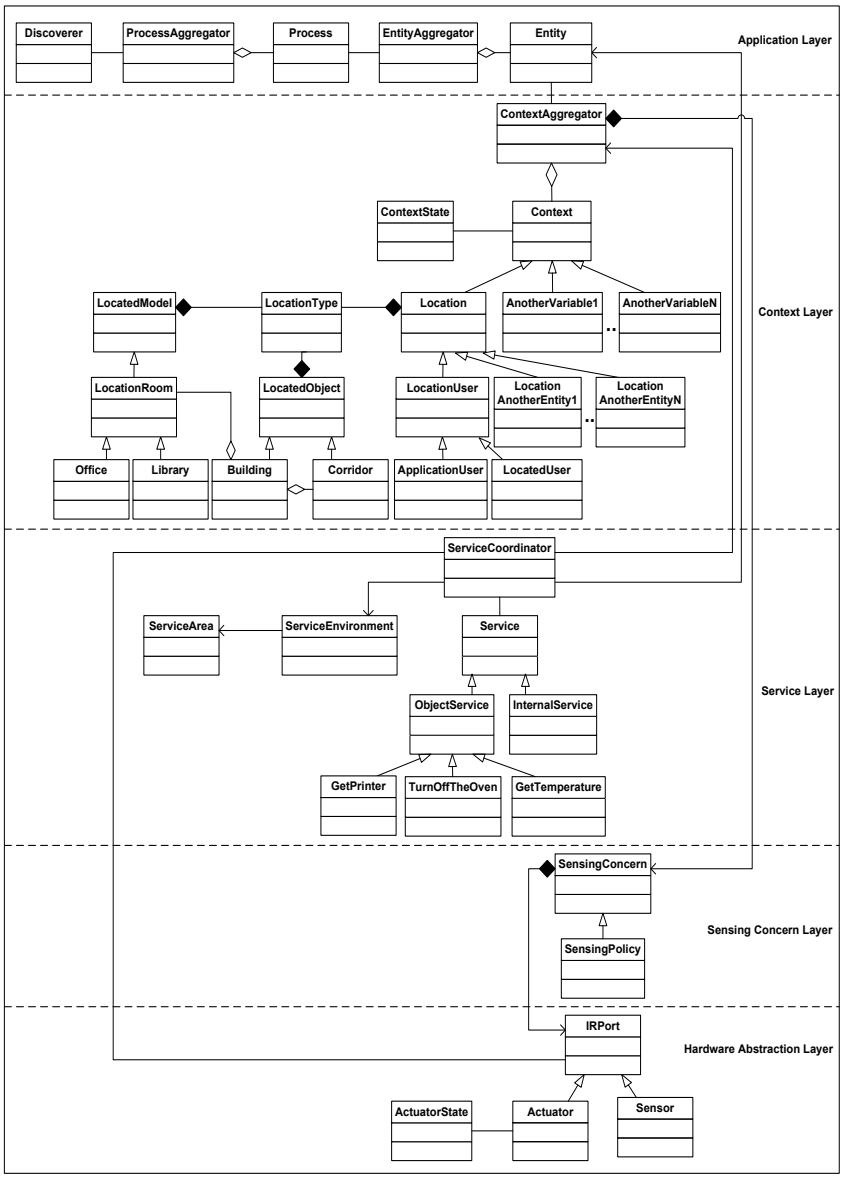


**Fig. 1.** Modelo MASCO

El Modelo procesa un cambio de valor de una o más variables de contexto, esto se refleja en la Clase IRPort, que se implementa como un patrón Observer. Un Objeto SensignConcern es notificado en la siguiente capa, para analizar y transformar las variables de contexto a valores comprensibles por la aplicación, esta clase se implementa como un patrón Strategy [5]. Posteriormente se notifica a la Clase ContextAggregator de la Context Layer, que se implementa como un patrón Observer

y Mediator [5], si la variable de contexto corresponde a la "ubicación" se la incorpora al objeto correspondiente en Context Layer, que es enviada a un objeto Location que representa la posición actual de la entidad, indicando un cambio de posición de las entidades sensadas.

La capa Service se encarga de la solicitud de servicios, donde ServiceCoordinator, constituye una interface común a todos los servicios, la Clase Service posee dos subclases: ObjectService e InternalService, la primera representa los servicios que son provistos a los objetos en general y la segunda es un servicio que es solicitado por otros servicios, como por ejemplo un cálculo intermedio. Para cada posible servicio hay una subclase concreta de ObjectService. Los servicios relacionados con ubicación, ServiceArea y ServiceEnvironment colaboran con ServiceCoordinator. ServiceArea tiene el conocimiento de la relación del objeto Location con los servicios que son provistos en el área, mientras que ServiceCoordinator conoce la relación con el área en la cual es localizado, por esto ServiceEnvironment actúa como un Patrón Mediador entre ServiceCoordinator y ServiceArea.

Los servicios solicitados por ServiceCoordinator se pueden brindar a entidades, manipulando los actuadores representados en Hardware Abstraction Layer por la clase Actuator, y su estado mediante ActuatorState, para esto se la implementa como un patrón State [5]. En Application Layer la clase Discoverer es responsable de mantiener la información sobre los procesos, entidades y variables de importancia en una aplicación. La clase EntityAggregator coordina la interacción entre objetos Entity, por esto, se implementa como un patrón Mediator [5], y se relaciona con un objeto Process, en caso de haber varios procesos. Cada uno de estos se corresponde a un objeto Discoverer (responsable del mantenimiento de la información sobre los objetos Entity y sus variables de contexto que trabajan en el proceso) y pueden estar relacionados con uno o más objetos Entity, además a cada uno de éstos se los puede relacionar con un objeto ContextAggregator. Cuando una variable de contexto cambia su valor EntityAggregator determina de qué manera deben relacionarse las variables restantes. Una vez definido el comportamiento, solicita a ServiceCoordinator la ejecución de un determinado servicio. Este servicio requiere la solicitud de un actuador, por lo que la subclase Actuator de IRPort contiene operaciones que hacen referencia a los servicios que brinda el objeto real.

## 4 Caso de Estudio

El caso de estudio corresponde a un robot de movimiento de avance, que recorre una habitación y detecta objetos a través del uso de un sensor sonar para determinar su ubicación. Ésta, está dada por los grados y la distancia a los que se encuentra el objeto en referencia a la ubicación actual del robot.

El robot realiza movimientos de avance, de giros: derecho e izquierdo, retroceso y se detiene. Además realiza la detección de un objeto luego, el robot se ubica de manera que el objeto es asido y depositado en un compartimento del mismo, cumpliendo su objetivo: ser un robot recolector.

Para determinar la ubicación de un objeto el robot posee un sensor sonar que a través de ondas sonoras permite determinar la distancia al objeto, además el sensor

recorre un espectro de 180°, los que se dividen en 4 secciones de 45° para determinar los grados a los que se encuentra (Fig.2).

Al determinar la presencia de un objeto, se procesa la información de su ubicación y se decide los servicios a ser ejecutados para realizar el agarre del mismo mediante un brazo robótico incluido como parte del robot (Fig. 3), formado por 4 secciones: hombro, brazo posterior, antebrazo y mano.

Los movimientos que realiza el robot son los siguientes (Fig. 4): Avanzar, Girar a la Izquierda, Girar a la Derecha, Detener, Retroceder.

Los movimientos que realiza el brazo robótico son los siguientes: a) Mover Hombro: Movimiento de rotación de hasta 180°; b) Mover Brazo Posterior: Movimiento de traslación de hasta 180°; c) Mover Antebrazo: Movimiento de traslación de hasta 160°; d) Movimiento de mano: Incluye movimiento de rotación de hasta 180° y de agarre.

Los movimientos del brazo robótico consisten en posicionar al mismo para que las pinzas (extremidades de la mano) puedan sujetar un objeto con características conocidas pero ubicación desconocida. [3]

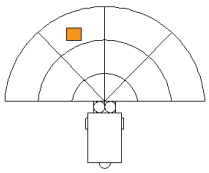

**Fig. 2.** Detección de Objetos mediante Sonar. [6]

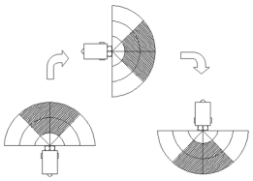

**Fig. 4.** Movimientos del robot. [6]

El brazo posterior, antebrazo y la distancia entre ambos forma un triángulo a partir del cual se puede definir la posición de ambos con respecto a los ángulos de traslación, para que la mano, que tiene una posición horizontal con respecto al eje x, pueda asir el objeto. (Fig. 5)

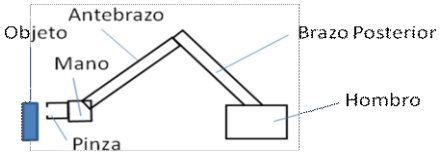


**Fig. 3.** Brazo Robótico.

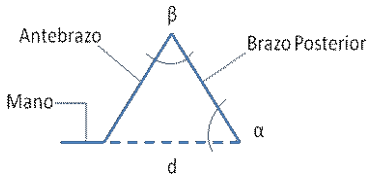


**Fig. 5.** Posición de agarre de un objeto.

La distancia desde el robot al objeto, definida como "D", se determina a través del sensado que realiza el sensor y procesamiento para convertir tiempo de retardo en esa distancia y los grados a los que se encuentra, sin embargo, se puede definir esta distancia como la suma entre el largo de la sección mano (que incluye a las pinzas de agarre) y una distancia d a determinar, como se indica en (1):

$$D = a + d \quad \textbf{(1)}$$

Siendo:

D: Distancia entre el robot y el objeto detectado, valor sensado.
a: longitud de la sección mano, valor conocido.

d: distancia entre el brazo y el objeto para realizar el agarre de un objeto.

Para calcular d, se obtiene (2) al despejar "d" de (1):

$$d = D - a \quad \textbf{(2)}$$

Dada las longitudes conocidas del antebrazo y brazo posterior y la distancia "d" calculada en (2), es posible determinar los grados α y β, que corresponden a los grados de giro de las dos secciones mencionadas, utilizando (3)

$$\text{Tan } \gamma = Co/Ca \quad \textbf{(3)}$$

Siendo:

γ: Ángulo a determinar.
Co: Cateto opuesto, valor conocido.
Ca: Cateto adyacente, valor conocido.

Así para calcular el ángulo de movimiento de traslación del brazo posterior se utiliza (4):

$$\alpha = \text{Arcotan } (l_a/d) \quad \textbf{(4)}$$

Siendo:

α: Ángulo de giro para realizar el movimiento de traslación del brazo posterior.
$l_a$: longitud de la sección antebrazo, valor conocido.
d: distancia entre el brazo y el objeto para realizar el agarre de un objeto. Calculada en (2)

Para el cálculo del ángulo de movimiento de traslación del antebrazo se utiliza (5):

$$\beta = \text{Arcotan } (d/l_b) \quad \textbf{(5)}$$

Siendo:

β:Ángulo de giro para realizar el movimiento de traslación del antebrazo.
$l_b$: longitud de la sección brazo posterior, valor conocido.
d: distancia entre el brazo y el objeto para realizar el agarre de un objeto. Calculada en (2)

### 4.1 Hardware Abstraction Layer

La Hardware Abstraction Layer (Fig. 6) consta de la Clase IPPort, que se implementa como un sensor o un actuador con un patrón Observer [5]. La Clase Actuator posee un estado, esto es, para que al momento de definir un servicio, éste ejecute una acción permitida de acuerdo al estado actual del actuador, mediante un patrón State [5]. Se definen las clases Port para representar la conexión con los servomotores físicos de un robot, donde cada servomotor se implementa como un objeto Servo que realizan movimientos de avance o de agarre. La Clase Sonar representa el sensor que detecta un objeto a través de la emisión de ondas sonoras. [6]

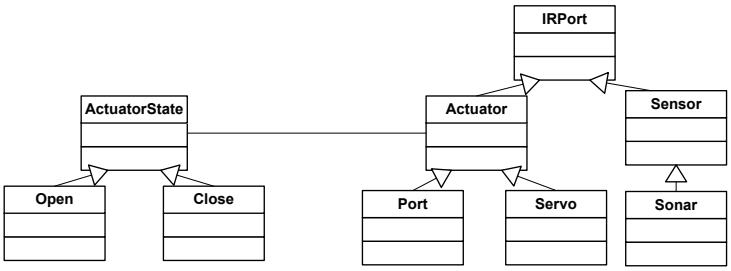


**Fig. 6.** Hardware Abstraction Layer adaptada al caso de estudio.

### 4.2 Sensing Concern Layer

La Sensing Concern Layer (Fig. 7) corresponde a la capa donde se realiza el procesamiento de las variables sensadas para producir un valor que pueda ser comprensible por las demás capas de la aplicación. La Clase SensingConcern se implementa como un patrón Strategy [5]. Se establece la clase LocationObjectPolicy que transforma el valor sensado en la distancia entre el objeto y el robot, como se indica:

$$D = V_s * T/2 \quad \textbf{(6)}$$

Siendo:

D: Distancia entre el robot y el objeto.
$V_s$: Velocidad del Sonido, conocida.
T: Tiempo de retardo. Valor sensado.

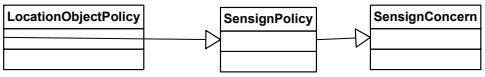


**Fig. 7.** Sensing Concern Layer adaptada al caso de estudio.

### 4.3 Context Layer

En la Context Layer (Fig. 8) se toma la decisión de qué servicios deben ser solicitados en la siguiente capa, Service, para esto la Clase ContextAggregator se implementa como un patrón Mediator [5]. Las clases Distance y Degree representan las dos variables de contexto del caso de estudio, ambas determinan la ubicación del objeto a ser asido, que se registra en el objeto InitialLocationObject. También se define la Clase RobotLocation [6] para definir la ubicación del robot dentro de la habitación. La Clase FinalLocationObject representa la ubicación final del objeto, en el compartimento del robot.

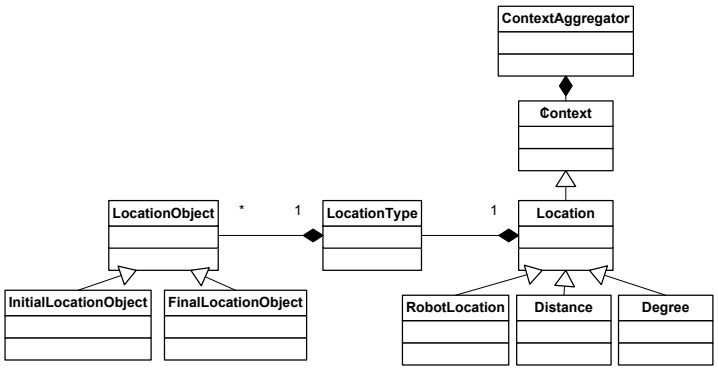


**Fig. 8.** Context Layer adaptada al caso de estudio.

### 4.4 Service Layer

La capa Service Layer (Fig. 9) posee la clase ServiceCoordinator que se implementa como un patrón Mediator [5], y coordina los servicios para realizar un movimiento del robot o el agarre de un objeto. La clase ObjectService establece los servicios de movimiento del robot: de giros hacia la izquierda y derecha, de movimientos adelante, atrás y de parada que consume el servicio interno de la Clase Motion, y de agarre que corresponde a los servicios para mover el brazo: MoveSoulder, MoveRearArm, MoveForeArm, MoreHand, HoldObject, DropObject, los que a su vez utilizan los servicios internos, que corresponden a las clases Traslation y Rotation.

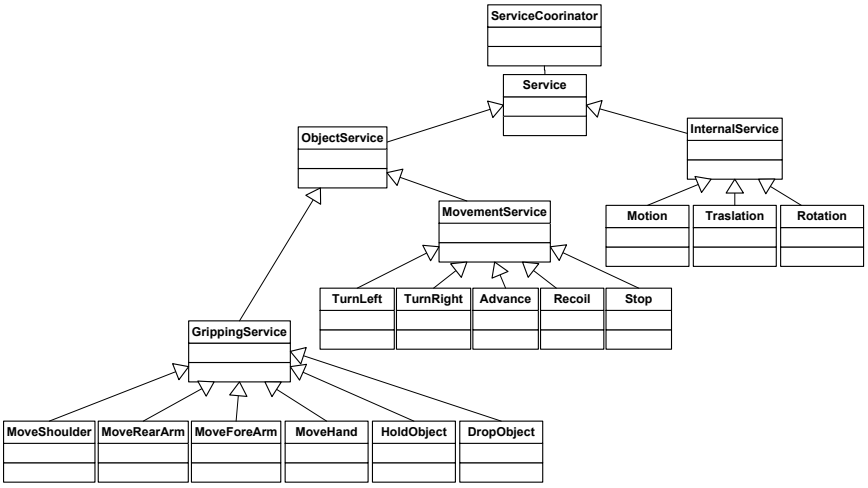


**Fig. 9.** Service Layer adaptada al caso de estudio.

### 4.5 Application Layer

La capa Application Layer (Fig. 10) gestiona la información de las entidades más importantes. Para el caso de estudio se representa el brazo robótico como una entidad, ya que posee secciones que deben coordinarse para realizar el agarre de un objeto. Se define una Clase por cada sección para su coordinación.

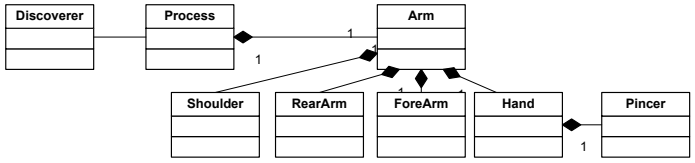


**Fig. 10.** Application Layer adaptada al caso de estudio.

Finalmente, el Modelo adaptado resultante (Fig. 11) contempla el movimiento de un robot, la detección y agarre de objetos.

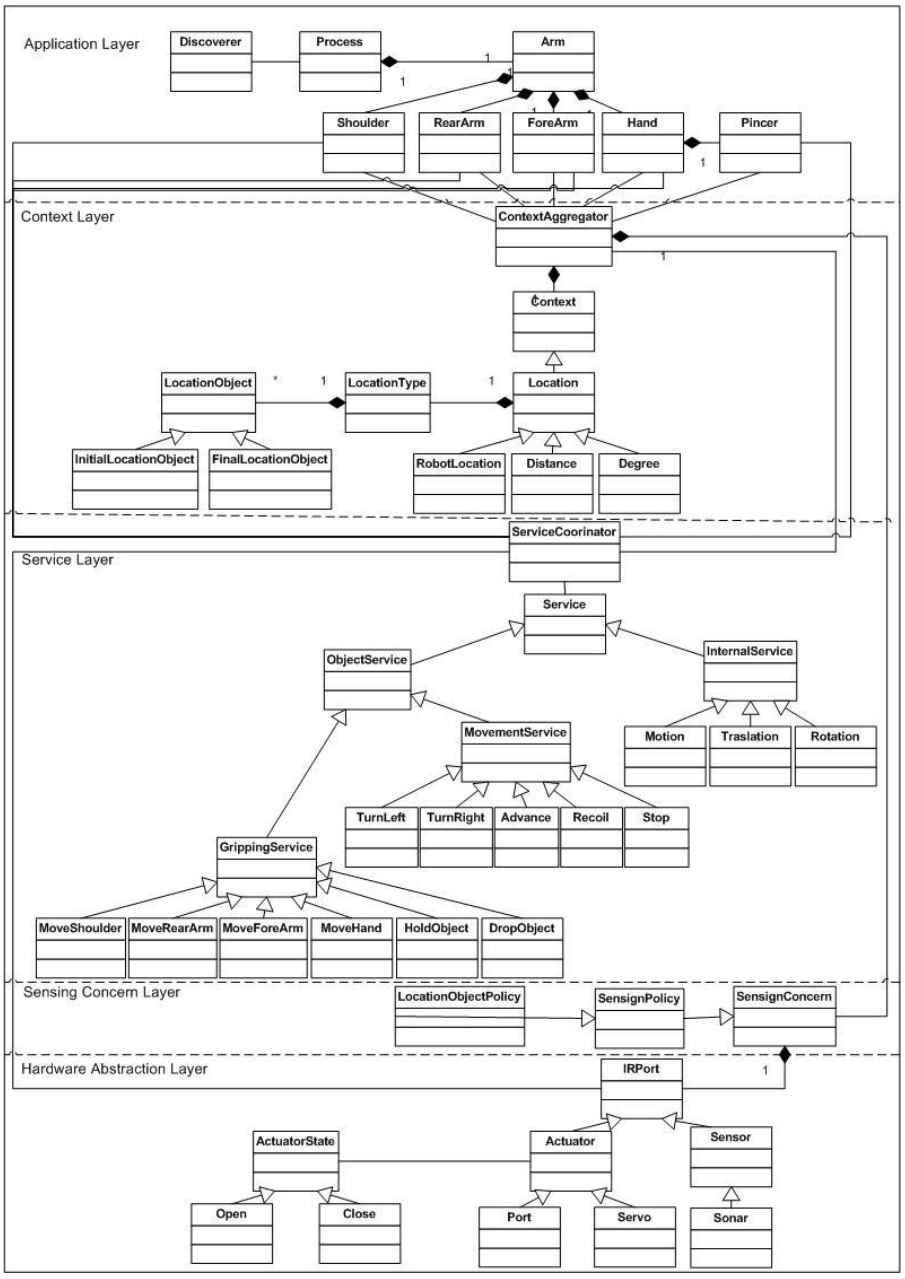


**Fig. 21.** Modelo MASCO adaptado al caso de estudio

## 5 Conclusiónes

El presente trabajo concluye con la adaptación del Modelo MASCO al caso de estudio: Robot de avance recolector de objetos, que se implementó utilizando un robot a escala con una placa Arduino Uno, manejado por un sistema sofware desarrollado en Visual Studio 2013 Community de licencia gratuita.

Para esto se adaptaron las capas de MASCO con Clases que representan sensores y actuadores necesarios para su funcionamiento, y también servicios para que el robot cumpla con su objetivo: asir objetos con una posición indefinida inicialmente, pero con un peso aceptable para posteriormente depositarlos en un compartimento propio. También permite el movimiento del robot en una habitación para encontrar estos objetos. Esto fue posible debido a que MASCO es un Modelo en capas, que posee las conexiones necesarias para que cada capa se conecte con aquellas que hacen un procesamiento necesario. Estas capas en general no son adyacentes para el proceso de decisión y ejecución de servicios, esto permite mayor rapidez en el funcionamiento del robot, esto es debido al uso de patrones en el diseño de MASCO.

Además al ser Orientado a Objetos es flexible para representar entidades como las secciones del brazo que forman parte del robot y las variables de contexto que determinan la posición de un objeto.

Se concluye que MASCO no sólo es aplicable al dominio de la Robótica como se demostró en [6] sino que además permite el manejo de entidades y varias variables de contexto, como así también servicios muy diferentes: los referidos a movimientos de avance y los de agarre, ampliando su utilización como una solución de Software.

## Referencias


1. http://robotica.wordpress.com/about/
2. Zabala, G.: Robótica, Guía Teórica y Práctica. User Power, Argentina (2011)
3. Velázquez, E.: Sistema de Comando de Movimiento de Brazo Robótico, Trabajo Final de Carrera. Facultad de Ingeniería, UNJU, Jujuy (2015)
4. Gálvez, M.P., Cáceres, N.R, Velázquez, C. E., Guzmán, A. N.: Atributos de calidad del modelo para aplicaciones sensibles al contexto MASCO. En: 5to Simposio Internacional de Investigación: "Interdisciplinariedad, Multidisciplinariedad y/o transdisciplinariedad: en la búsqueda de respuestas desde las experiencias de investicación". Universidad Católica de Santiago del Estero, DASS, Jujuy (2013)
5. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable OO Software. Addison Wesley, USA (1995)
6. Velázquez, E., Guzmán, A., Gálvez, M.P.: Modelo en capas sensibles al contexto aplicada a la movilidad de un robot. En: Cuarto Simposio Internacional de Investigación. Libro de Resúmenes. Pág. 117. Universidad Católica de Santiago del Estero, DASS, Jujuy (2011)