



UNIVERSIDAD DE CHILE
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

Diseño e Implementación de Memoria de Largo Plazo para Robots de Servicio

PROPUESTA DE MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO E INGENIERO CIVIL EN COMPUTACIÓN.

Autor:

Matías PAVEZ

matias.pavez@ing.uchile.cl

+569 9888 9358

Profesor Guía: (DIE)

Javier RUIZ DEL SOLAR

jruizd@ing.uchile.cl

Co-Guía: (DCC)

Jocelyn SIMMONDS

jsimmond@dcc.uchile.cl

Santiago de Chile
Junio, 2017

Índice general

1. Introducción	3
1.1. Antecedentes Generales	3
1.1.1. Robots de servicio domésticos	3
1.1.2. Equipo de Trabajo: UChile Homebreakers	4
1.1.3. La memoria humana	4
1.2. Motivación	5
1.2.1. Problema	5
1.2.2. Oportunidad	6
1.3. Objetivos	7
1.3.1. Objetivo General	7
1.3.2. Objetivos Específicos	7
1.4. Alcances y Contribución del Trabajo	8
2. La Memoria y Robótica	9
2.1. Robots de Servicio Domésticos	9
2.2. Memoria Humana	10
2.2.1. Memoria de Corto Plazo	12
2.2.2. Memoria de Largo Plazo	12
2.2.3. Plasticidad Sináptica y Modulación	13
2.3. Memoria y robótica	14
2.3.1. Relevancia de la memoria robótica	14
2.3.2. Relación entre memoria humana y memoria robótica	15
2.3.3. Memoria LTM explícita	16

<i>ÍNDICE GENERAL</i>	2
2.3.4. Memoria Emocional	18
2.3.5. Otros enfoques	20
3. Aspectos Técnicos	21
3.1. ROS	21
3.2. UChile ROS Framework	22
3.3. KnowRob	23
3.4. Plataforma objetivo: Robot Bender	23
4. Metodología	25
4.1. Propuesta de Solución	25
4.2. Planificación	26

Capítulo 1

Introducción

En este capítulo se presenta el trabajo de título a realizar. Se da el contexto en el que se enmarca, la motivación para su desarrollo y los objetivos del proyecto. Se revisan los conceptos de robótica doméstica, la memoria humana y el equipo de trabajo donde se implantará el software. Luego se detallan los problemas actuales y la oportunidad de desarrollo. Finalmente se formalizan los objetivos del trabajo así como sus alcances.

1.1. Antecedentes Generales

A continuación, el lector encontrará una breve introducción a los temas requeridos para contextualizar este trabajo: La robótica de servicio doméstica y el equipo de trabajo donde se implantará la solución. Además, se introduce el tema de la memoria humana, requerido para entender la propuesta y su relación con la robótica. Todos estos temas serán tratados en mayor profundidad más adelante, en los Capítulos 2 y 3.

1.1.1. Robots de servicio domésticos

La robótica de servicio es un área enfocada en asistir a los seres humanos en tareas repetitivas y comunes, como la recolección de basura. Para completar una tarea, el robot requiere cierto grado de autonomía, que le permita actuar en ambientes no controlados y utilizando sus sensores para responder correctamente a los cambios del entorno.

A grandes rasgos, los robots de servicio se categorizan en robots para el transporte, seguridad y domésticos. Los robots de servicio domésticos se caracterizan por realizar tareas de asistencia en el hogar y de compañía para humanos. Algunas de las tareas típicas que deben realizar son ayudar a ordenar, preparar comida u ofrecer bebestibles. Algunos se enfocan en el cuidado de adultos mayores, en mascotas de compañía, salud o educación.

1.1.2. Equipo de Trabajo: UChile Homebreakers

El laboratorio de robótica del Departamento de Ingeniería Eléctrica de la Universidad de Chile alberga dos equipos de robótica: *UChile Robotics Team*, dedicado al fútbol robótico y *UChile Homebreakers Team*, enfocado en robótica de servicio. Ambos son conformados por alumnos de pregrado y postgrado de diversas especialidades, y liderados por el profesor Javier Ruiz del Solar [1].

UChile Homebreakers existe desde el año 2007 y actualmente cuenta con 15 estudiantes. Todo su desarrollo de software está basado en ROS, un framework para el desarrollo de plataformas robóticas y con miles de usuarios alrededor del mundo [2].

El equipo trabaja en dos plataformas humanoides de tipo doméstico, Bender y Pepper. Bender es un robot construido en el mismo laboratorio y con el objetivo de ser un mayordomo para el hogar. Pepper, desarrollado por SoftBank Robotics [3], está diseñado para ser un robot de compañía. Ambos comparten la misma arquitectura de software y prácticamente todo su código, exceptuando los drivers para acceder al hardware respectivo.

RoboCup @Home League

La RoboCup es una competencia internacional cuyo objetivo es ser un vehículo para el desarrollo de la robótica y la inteligencia artificial. Está compuesta de variadas ligas: Rescue, Soccer, Simulation, @Home, Industrial y Junior, cada una con diversas subligas orientadas a fomentar la investigación de distintos aspectos del campo. Su sueño es que para mediados del siglo 21, un equipo de fútbol robótico completamente autónomo sea capaz de vencer al campeón de la última copa mundial y siguiendo las reglas de la FIFA [4].

UChile Homebreakers participa desde el año 2007 en la categoría @Home. Las pruebas de la liga se desarrollan en escenarios que imitan ambientes reales, como un hogar o un restaurante. Las capacidades generalmente evaluadas y potenciadas en @Home son de Visión Computacional, Navegación autónoma, Manipulación de objetos y Reconocimiento de Voz. Cada año el equipo planifica sus desarrollos de acuerdo a los requerimientos de la competencia, por lo que trabajos fuera de las áreas mencionadas no son considerados una prioridad.

1.1.3. La memoria humana

Según Eichenbaum [5], la memoria hace relación al almacenamiento de experiencias en el cerebro. Se pueden distinguir múltiples sistemas de memoria independientes y sustentados por distintas estructuras cerebrales. A grandes rasgos, la memoria se puede dividir en de corto plazo STM (Short-Term Memory) y de largo plazo LTM (Long-Term Memory). La STM maneja información muy detallada, es de poca capacidad y permite un rápido acceso, mientras que la LTM maneja mucha información sobre experiencias y entidades, es menos detallada y de acceso más lento.

Eichenbaum divide la LTM en una componente explícita (consciente) y una implícita (inconsciente). La primera almacena datos episódicos, pudiendo responder las preguntas “Qué”, “Dónde” y “Cuándo”, datos semánticos, que modelan hechos y conceptos como el lenguaje o personas, y también, las conexiones entre ambas submemorias. La memoria implícita codifica habilidades, hábitos y preferencias.

Existen procesos de consolidación y deterioro de la memoria que están constantemente en funcionamiento. La consolidación requiere un estímulo relevante, sumado al proceso de almacenamiento, lo que genera conexiones entre la memoria episódica y la respectiva zona semántica. En caso de haber experiencias repetidas, las conexiones se fortalecen. El deterioro de la memoria es un proceso que degenera las conexiones entre ambas formas de memorias explícitas.

La memoria emocional es una forma de memoria implícita que genera reacciones emocionales y sentimientos. Según los estímulos a los que se enfrente, permite modular el proceso de consolidación de la STM en LTM, modificando el nivel de relevancia de los eventos, pudiendo generar memorias muy fuertes y hábitos arraigados. Ejemplos de esto son los flashbacks y las memorias asociadas a eventos importantes.

1.2. Motivación

La memoria es una habilidad cognitiva crucial para los humanos. Al interactuar con otras personas o el ambiente les permite recordar experiencias pasadas y sus detalles. Luego, es de esperar que un robot de servicio posea una memoria que le permita potenciar sus capacidades de interacción con los humanos que ayudará [6]. Una LTM permitiría, por ejemplo, generar diálogos interesantes sobre eventos pasados o cosas que el robot puede inferir del comportamiento humano, por otro lado, también permitiría la generalización de las tareas que tiene que llevar a cabo.

Particularmente, dado el enfoque de las plataformas disponibles, Bender cómo robot mayordomo y Pepper cómo robot social, se espera que ambos posean capacidades avanzadas de interacción con los humanos, para lo que se requiere una LTM.

1.2.1. Problema

El año 2015 se desarrolló una LTM episódica para el robot Bender, orientada a la interacción con personas y objetos [7]. El trabajo consideraba métodos para almacenar, adquirir y manejar la información episódica, sumado a un proceso simple de consolidación de memoria.

Actualmente la memoria desarrollada no está operativa, ni es factible habilitarla. A continuación se listan los aspectos que se consideran causas del problema desde un punto de vista técnico y humano:

- No se integró adecuadamente al software del robot, no se recopila ni provee información continuamente mientras el robot está en funcionamiento.
- La memoria no provee una API que siga el estándar de los desarrollos del equipo, por lo que no se usa ni es mantenida.
- RoboCup@Home no considera el uso de LTM en sus competencias, por lo que el equipo no tiene un incentivo real para seguir desarrollando o mantener la memoria. Esto además ha provocado que el código quede obsoleto.

Por otro lado, suponiendo que lo anterior estuviese solucionado, aún existen los siguientes problemas:

- Sólo considera 2 modelos semánticos: Persona y Objeto, para los cuales sólo se almacena información de nombre, nacionalidad e imagen.
- A pesar de considerar un modelo para objetos, no se integró con los módulos relacionados que recopilan la información, por lo que realmente la memoria sólo funciona para entidades de tipo Persona.
- Es esperable que una memoria considere más modelos (como adultos, niños, animales u objetos) y más características para cada uno de los modelos (como nombre, hobbies, trabajo o edad, para el caso específico de un humano).
- La consolidación de memoria STM a LTM sólo considera la primera interacción con cada entidad, por lo que no existe actualización de los datos.
- Hay una restricción en los modelos y características a almacenar, respecto a la información que el robot es realmente capaz de obtener.

1.2.2. Oportunidad

Existe un vasto desarrollo respecto a la memoria y los procesos cognitivos, sin embargo, la investigación se concentra en campos como psicología, neurología y ciencias cognitivas. Los estudios de LTM para robots de servicio son muy acotados y no existe una solución estándar a implementar. Algunos robots, como la versión comercial de Pepper, utilizan LTM, pero el código asociado no es libre, ni está basado en ROS.

El uso de LTM no está en las prioridades “RoboCup” del equipo, sino que es algo útil para demostraciones y para potenciar la interacción humano-robot. Por ello, se considera que no basta con desarrollar un módulo capaz de recopilar información inteligentemente, sino que además se requiere una integración con las capacidades de diálogo o de inferencia de información, para finalmente proveer una demostración de estas habilidades.

Así, esta es una oportunidad para diseñar una LTM para robots de servicio, que considere aspectos como:

- Memoria episódica y semántica adecuada a tareas generales de robots de servicio.
- Metodología para consolidación de STM en LTM.
- Servicio para recopilación continua de información.
- Implementación basada en el framework ROS, siguiendo la línea de los desarrollos en UChile Homebreakers.
- Memoria emocional que permita dar relevancia a los eventos.
- Inferencia de información a partir de datos de la memoria. Por ejemplo: “Juan suele desayunar a las 9 am”, “El control de la TV suele estar en el sofá”, etc.

Tanto la memoria emocional como la inferencia de información se consideran requisitos deseables, por lo que están fuera del *core* del proyecto.

Más adelante se detallan cada uno de los aspectos anteriores. En el Capítulo 2 se hace una revisión de cada punto, respecto al estado del arte. Mientras que en el Capítulo 4 se propone una alternativa de solución a cada uno de ellos.

1.3. Objetivos

1.3.1. Objetivo General

El objetivo general es el diseño de una LTM para robots de servicio domésticos, que considere componentes episódicos y semánticos. La LTM debe ser integrada en Bender, recopilando recuerdos constantemente y con una API acorde a los desarrollos de UChile Homebreakers. Además, se debe proveer una demostración de las funcionalidades introducidas.

1.3.2. Objetivos Específicos

A continuación se presentan los objetivos específicos del trabajo, a modo de desglose del objetivo general en tareas más acotadas.

- Definir consultas para validación del sistema.
- Diseñar el proceso de consolidación de recuerdos.
- Diseñar la arquitectura del sistema.
- Implementar la LTM y su API.
- Implementar servicio que recopile recuerdos constantemente.
- Implementar la demostración.

1.4. Alcances y Contribución del Trabajo

Se espera que este trabajo de título sirva como base para el desarrollo de funcionalidades más avanzadas, basadas en memorias LTM en el futuro. Por lo tanto, el foco del trabajo será el diseño de la LTM y la implementación del software que soporte el diseño. La demostración sólo será utilizada como medio de validación del trabajo. Entonces, la principal contribución del trabajo es el diseño de la LTM para Bender.

Tanto la memoria emocional, como la inferencia de información se consideran objetivos secundarios, que no son del *core* del proyecto. Sin embargo, dado que son conceptos casi imprescindibles para una LTM, deben ser considerados en el diseño, a pesar de que no sean implementados.

En términos del desafío del trabajo, y tras una revisión del estado del arte y el software disponible (Capítulos 2 y 3), se cree que la mayor parte del esfuerzo estará en el diseño de una LTM Episódica compatible con URF y KnowRob. Tanto URF como KnowRob son revisados en el Capítulo 3. URF hace referencia a todo el software actualmente funcionando en el robot Bender. Se propone el uso del software KnowRob como base para la implementación de la memoria, pues está diseñado para proveer S-LTM, P-LTM e inferencias sobre ambas memorias.

Capítulo 2

La Memoria y Robótica

En este capítulo se revisan los temas y conceptos relevantes para el desarrollo del trabajo. Se formaliza la definición de robot doméstico y sus alcances. Se describe la memoria humana, sus categorías, funcionamiento y procesos cerebrales relevantes. Basando en los temas anteriores, se revisa la relación entre robótica y la memoria humana: se describen los distintos enfoques existentes, las reglas generales para su implementación y los frameworks actuales.

2.1. Robots de Servicio Domésticos

La Federación Internacional de Robótica (IFR) [8] define *robot* como:

“Un mecanismo actuado y programable en dos o más ejes y con un cierto grado de autonomía, que se mueve en su entorno para realizar tareas previstas. En este contexto, autonomía se refiere a la habilidad de realizar tareas previstas, basado en el estado actual y lo sentido, sin intervención humana.”

Asimismo, la IFR define un *robot de servicio* como un robot “que realiza tareas útiles para humanos o equipamiento, excluyendo aplicaciones de automatización industrial”. Así, un robot de servicio debe trabajar en ambientes no controlados y con la autonomía suficiente que le permita llevar a cabo su cometido. Generalmente, la robótica de servicio se enfoca en asistir a los seres humanos en tareas repetitivas y comunes.

Según su área de aplicación, un robot de servicio se clasifica en *de uso personal* o *de uso profesional*. Los primeros son utilizados en ambientes no comerciales y por personas comunes; como por ejemplo, un robot sirviente o una silla de ruedas autónoma. Un robot de servicio profesional se utiliza en ambientes comerciales, usualmente operados por alguien entrenado; un ejemplo son los robots de entrega de paquetes o para cirugía.

Según la recopilación de datos realizada por la IFR durante el 2016, este tipo de robots es utilizado en las siguientes áreas:

- Tareas domésticas: De compañía, asistencia, limpieza, cuidado del hogar.
- Entretenimiento: Juguetes, comunicación, educación e investigación.
- Asistencia a ancianos y discapacitados: Sillas robóticas y robots para cuidar personas.
- Transporte.
- Seguridad y vigilancia.
- Otros que no caen en las categorías anteriores.

El foco de este trabajo son los robots de servicio personales, dedicados a tareas domésticas, clasificación a la que en adelante se referirá como *Robots Domésticos*.

Para entender el alcance del trabajo, en cuanto a qué es lo que se espera del sistema, a continuación se listan algunas capacidades de los robots domésticos. Un robot de compañía y asistencia tiene, pero no se limita a las siguientes tareas:

- Interacción amistosa con humanos.
- Ayudar a recordar y organizar tareas.
- Cooperar con la realización de un procedimiento.
- Guiar y seguir a personas.
- Recordar información y entidades.

Algunas tareas que robots domésticos de tipo mayordomo deben ejecutar son:

- Ofrecer comida y bebestibles.
- Preparación de comida.
- Ordenar y limpiar el hogar.

El primero de los objetivos específicos del proyecto tiene que ver con la derivación de consultas útiles para validar la memoria, a partir de las capacidades esperadas para un robot de servicio doméstico. En el Capítulo 4 se presenta cómo se planea desarrollar esta tarea.

2.2. Memoria Humana

La memoria es un elemento fundamental para los humanos en su día a día, es parte integral de su existencia. Permite recordar quién, qué, cómo, dónde y cuándo. En términos psicológicos, es la habilidad para para codificar, almacenar y luego obtener información sobre eventos pasados, en el cerebro. Los pensamientos son parte de la memoria de corto plazo, mientras que eventos pasados son almacenados en una memoria de largo plazo. Existen

muchos estudios en el área de la psicología cognitiva con diversas descripciones y modelos teóricos de cada tipo de memoria [6].

Desde el punto de vista de la información procesada, la memoria es vista como una facultad humana consistente en procesos para el manejo de información. Los 3 componentes principales son:

- **Codificación:** En este paso, se adquiere nueva información desde los sentidos humanos. Los datos son convertidos a un formato que pueda ser almacenado en la estructura cerebral correspondiente.
- **Almacenamiento:** Consiste en la creación de registros permanentes de información. Es un proceso pasivo, de continuo procesamiento para clasificar datos nuevos y los ya existentes en el cerebro.
- **Adquisición:** Hace referencia al acceso de datos almacenados. El proceso se realiza en el momento, para obtener una reconstrucción aproximada de la información, a partir de elementos repartidos en distintas partes del cerebro.

La memoria puede ser dividida en múltiples sistemas de independientes, con funcionalidades bien definidas y sustentados por distintas estructuras cerebrales. La primera diferenciación define dos tipos de memoria: la memoria de corto y la de largo plazo, STM (Short-Term Memory) y LTM (Long-Term Memory), por sus siglas en inglés. En el diagrama de la Figura 2.1 se muestra una separación clásica utilizada en el área de las ciencias cognitivas [5], explicada en las siguientes subsecciones.

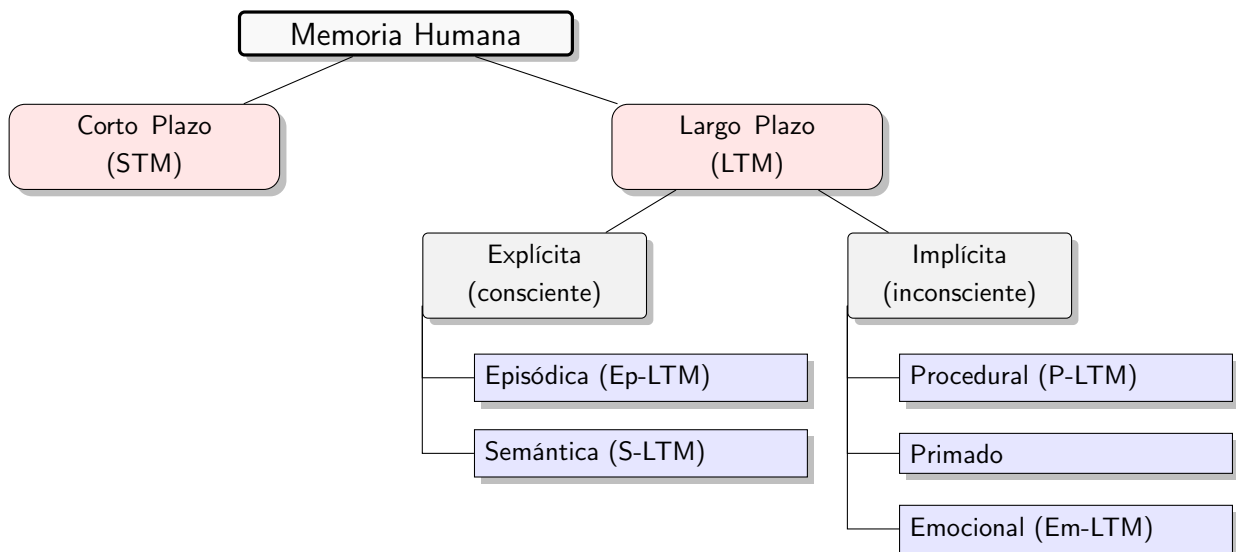


Figura 2.1: Clasificaciones de la memoria humana.

2.2.1. Memoria de Corto Plazo

En el ámbito cognitivo, la STM se refiere a la habilidad de estar atento, recopilar información y memorias, para luego utilizarlas dentro de un corto periodo de tiempo. Es responsable de almacenar información constantemente y de decidir que parte será transferida a la memoria de largo plazo. El término de *Memoria de Trabajo* suele ser utilizado de manera intercambiable con el de STM.

La STM se caracteriza por manejar información muy detallada, ser de poca capacidad y permitir un rápido acceso a estos datos. Permite recordar rápidamente y con gran detalle experiencias ocurridas hace pocos segundos, pero con dificultad creciente a medida que avanza el tiempo.

Se sustenta principalmente en la corteza prefrontal del cerebro. Algunos estudios han mostrado que las neuronas involucradas son capaces de mantener información relevante de corto plazo, la que es combinada con información sensorial entrante y áreas que manejan la toma de decisiones. En los humanos esta área presenta gran activación durante procesos de codificación, acceso y manipulación de memorias.

2.2.2. Memoria de Largo Plazo

La LTM se asocia al almacenamiento permanente de información en el cerebro. Se caracteriza por manejar mucha información sobre experiencias y entidades, ser menos detallada y proveer un acceso más lento a los recuerdos, respecto a la STM [5]. Cierta información de la STM eventualmente es transferida a la LTM. De acuerdo a la Figura 2.1, sus dos principales categorías son la *Memoria Implícita* y la *Memoria Explícita*.

Memoria Explícita

La memoria explícita suele ser denominada *memoria consciente* o *memoria declarativa*, pues maneja conocimientos relacionados a hechos y eventos adquiridos de forma consciente. Según las estructuras cerebrales involucradas, se conforma de la *memoria episódica* y de la *memoria semántica*.

La memoria episódica (Ep-LTM) es de carácter autobiográfico y almacena detalles de eventos y experiencias pasadas. Permite responder a las preguntas “Qué sucedió”, “Dónde ocurrió” y “Cuándo ocurrió”. Un humano puede acceder a esta memoria si es capaz de decir: “recuerdo que”. Este tipo de memoria da al ser humano la sensación de continuidad en el tiempo.

La memoria semántica (S-LTM) almacena el conocimiento de hechos, significados, categorías y proposiciones. Un humano puede acceder a esta memoria si es capaz de decir: “sé que”. Esta memoria se abstrae de perspectiva e información situacional.

Las estructuras cerebrales que soportan la memoria explícita son el hipocampo, encargado

de manejar la Ep-LTM, junto a la corteza cerebral, en donde se distribuyen los conocimientos de la S-LTM. En el hipocampo se mantienen conexiones neuronales a los sectores de interés de la corteza, en donde se alojan conocimientos semánticos asociados a cada episodio.

Un ejemplo de uso de Ep-LTM es el recuerdo de una graduación escolar, el lugar y la fecha donde ocurrió. La S-LTM podría responder en que consiste una graduación y describir la ropa que se suele ocupar en ellas.

Memoria Implícita

La memoria implícita abarca la capacidad de aprender habilidades, hábitos y preferencias, caracterizados por ser mejorados o adquiridos sin una recolección consciente. Así, también suele ser denominada *memoria inconsciente* o *memoria no declarativa*, pues comprende acciones que pueden ser realizadas sin pensar en ellas. Ejemplos de esto, son el andar en bicicleta o tocar un instrumento musical.

Dos de sus componentes son la *memoria procedural* (P-LTM) y la *memoria de primado*. La primera ayuda a realizar tareas sin pensar en ellas, es decir, maneja el conocimiento del *Cómo*; Ejemplos de esto son comer y caminar. La memoria de primado hace referencia a la predisposición para recordar hechos o información a la que un sujeto es expuesto con anterioridad; Ejemplos de esto son la facilidad para recordar canciones escuchadas hace poco tiempo, o el uso de palabras e ideas vistas recientemente.

Se ha mostrado que la P-LTM se sustenta en el cerebelo, mediante la activación de este durante el uso de habilidades motoras.

Un tercer componente de la memoria implícita es la *memoria emocional* (Em-LTM). Se encarga de dar significado afectivo a ciertos estímulos, que de otra forma serían neutrales. Las estructuras cerebrales involucradas son la amígdala, áreas corticales y subcorticales. Esta memoria se expresa mediante la activación del hipotálamo, en conjunto al sistema nervioso simpático, generando reacciones emocionales y sentimientos.

2.2.3. Plasticidad Sináptica y Modulación

Se denomina *consolidación* de memoria al proceso de transición de conocimiento desde la STM a la LTM. Durante la consolidación se generan conexiones neuronales entre la Ep-LTM y la respectiva zona semántica. Para activar la consolidación se requiere de un estímulo relevante, sumado a la cadena de eventos para el almacenamiento.

Se denomina *deterioro* de memoria u “olvido” al proceso de debilitamiento de las conexiones neuronales establecidas por los procesos de consolidación. Está en constante funcionamiento, degenerando las asociaciones entre la Ep-LTM y la S-LTM. Por lo tanto, en este contexto, el olvido no significa una eliminación de los datos en el cerebro, sino que estos siguen ahí, pero la conexión requerida es inexistente o es demasiado débil para poder ocuparla.

Existen procesos químicos a nivel cerebral que afectan la consolidación y el deterioro de la LTM. Hay evidencia de que estos están en continuo funcionamiento. Estos eventos celulares ocurren en una escala de segundos a minutos, y son esenciales para la mantención de la memoria a largo plazo.

Es posible modular ambos procesos. Las experiencias repetidas potencian la consolidación de la memoria, lo que fortalece las conexiones neuronales. Por otro lado, la memoria emocional es capaz de potenciar o deprimir las reacciones químicas requeridas; según los estímulos a los que se enfrente, modifica el nivel de relevancia de los eventos, pudiendo generar memorias muy fuertes y hábitos arraigados. Ejemplos de esto, son la memorización por repetición, los flashbacks y las memorias asociadas a eventos importantes como cumpleaños.

2.3. Memoria y robótica

En esta sección se presenta el estado del arte respecto al uso de LTM en robótica. En primer lugar se presenta la importancia de la LTM y las expectativas para un robot doméstico. Luego se hace una comparación entre la memoria humana y el manejo de información en robots. Se presenta el estado del arte para cada componente cognitivo de interés. Finalmente, se describen otros enfoques de la literatura para la implementación de LTMs, que no están basados en el enfoque biológico utilizado en este proyecto.

2.3.1. Relevancia de la memoria robótica

La memoria es una habilidad esencial para cualquier ser social. Lo mismo aplica para un robot doméstico cuya misión sea establecer una relación de largo plazo con usuarios humanos. Un problema común es que los usuarios tienden a perder el interés rápidamente en los robots, debido a la falta de vida y expectativas no cumplidas, respecto a la inteligencia y capacidad de socializar de la máquina. El problema se potencia con el paso del tiempo, donde la motivación por interactuar disminuye y se genera frustración, a medida que el robot continua repitiendo los mismos comportamientos predefinidos [9].

Si se desea mejorar la interacción humano-robot, entonces se requiere que el robot se comporte de manera más natural. Los mejores agentes robóticos sociales deberían satisfacer las necesidades cognitivas y sociales humanas; mientras más familiar sea la interacción, serán más efectivos en su propósito. Así, la LTM es una habilidad crucial si se espera que el robot sea capaz de aprender y adaptarse a su entorno.

Por otro lado, desde un punto de vista práctico, se ha mostrado que el concepto de memoria LTM aplicada a robots es beneficioso. En [10] ocupan memoria P-LTM para mejorar el desempeño de un robot en ambientes dinámicos, logrando acelerar el proceso de adaptación al entorno y la toma de decisiones.

2.3.2. Relación entre memoria humana y memoria robótica

Son muchos los trabajos en LTM que han basado su desarrollo en la taxonomía de la memoria humana, donde se implementan esquemas de información con módulos análogos a los presentados en la Figura 2.1. Esto se puede justificar por la similitud de cada tipo de memoria, con módulos preexistentes en la arquitectura robótica. A continuación se presenta una comparación entre cada tipo de memoria, sus procesos y el análogo robótico.

Memoria STM

Se relaciona a todos los datos que están actualmente cargados en la memoria primaria de la máquina. Esta memoria es la utilizada para solucionar la tarea actual, es equivalente a los pensamientos del robot y cumple con las características de la STM humana: es volátil, de rápido acceso y limitada en capacidad. También se encuentra presente en todo archivo temporal manejado por el sistema, mientras está en funcionamiento. Así, la estructura equivalente a la cerebral sería principalmente la RAM de la máquina.

Memoria S-LTM

La memoria S-LTM es común y se puede asociar a casi toda fuente de datos estática, no utilizada por las otras memorias. Luego, la S-LTM se sustenta en la memoria secundaria, cumpliendo las características de la LTM humana: es persistente, de acceso costoso y virtualmente ilimitada en capacidad. Algunos ejemplos son:

- Bases de datos.
- Directorios con imágenes de personas y objetos conocidos.
- Mapa con descripción del ambiente.
- Frases predefinidas que puede decir el robot.
- Archivos de audio utilizados por el robot.
- En general, todo archivo con datos persistentes, cargados en cada sesión de trabajo.

Memoria P-LTM

Este tipo de memoria es comparable a algoritmos predefinidos para realizar acciones, generalmente motoras.

Algunos ejemplos comparables son:

- Algoritmos basados en redes neuronales, entrenados para manipular objetos o reconocer patrones.

- Algoritmos entrenados para tareas específicas, cómo la detección de caras o el reconocimiento de voz.
- Controladores basados en puntos de operación para acciones motoras.
- Síntesis de voz.

Las estructuras equivalentes a la versión cerebral serían los archivos con parámetros para cada algoritmo, obtenidos a partir del entrenamiento o ajustados manualmente.

Otros tipos de memoria

Generalmente, tanto STM, S-LTM como P-LTM son un requisito mínimo para el funcionamiento de un software robótico, por lo que no son implementadas de forma explícita, sino que se pueden identificar en los componentes de software descritos anteriormente. Luego, la existencia de tales memorias, no implica la intención de crear una arquitectura LTM similar a la humana. Los otros tipos de memorias sólo son implementados en casos especializados.

2.3.3. Memoria LTM explícita

La memoria de mayor interés para este proyecto es la Ep-LTM, pues es la que permite generar interacciones humano-robot interesantes y que no sean repetitivas en el tiempo. Para su implementación, es un requisito disponer de S-LTM, pues la Ep-LTM almacena episodios y los cambios ocurridos a las entidades percibidas. La S-LTM almacena los modelos de cada entidad que llegarán a utilizar por la Ep-LTM.

No existe un consenso sobre los contenidos, el formato o las herramientas para implementar una Ep-LTM. Sin embargo, si existe una aceptación generalizada sobre los requerimientos mínimos y deseables para el diseño [6, 9, 11, 12]:

Aspectos de diseño requeridos:

1. Contenido: (R1) La información de eventos pasados debe ser recolectada e indexada respecto a su contexto espacio-temporal: Qué, dónde y cuándo pasó.
2. Estructura: (R2) Cada evento en conjunto con su contexto espacio-temporal forman una única representación integrada, que debe ser recordada como un todo, en caso de obtener cualquiera de las características del evento.
3. Flexibilidad: (R3) La información almacenada es declarativa por naturaleza, y puede ser flexiblemente almacenada. Particularmente, puede interactuar con conocimiento semántico, incluso si este fue obtenido con posterioridad a la codificación del episodio.
4. Datos específicos: (R4) La memoria episódica cuenta con sólo una instancia de cada evento para su entrenamiento, pues cada evento tiene características específicas a la situación.

5. Ep-LTM es LTM y declarativa: (R5) La memoria episódica es una forma de memoria LTM. Puede almacenar recuerdos por segundos, minutos, días o años. También es una forma de memoria declarativa; Es posible hablar sobre eventos asociados y acceder a ellos para introspección.
6. Perspectiva: (R6) La memoria episódica debe lidiar con datos específicos al evento, lo que implica una perspectiva. Es decir, eventos recordados deben mantener la misma perspectiva que se tenía en la experiencia original.
7. Anidamiento: (R7) Los eventos almacenados en la memoria episódica pueden variar en tiempo y extensión. Particularmente, pueden ocurrir eventos dentro del actual.
8. Trasposición : (R8) Los eventos almacenados en la memoria episódica pueden variar en tiempo y extensión. Particularmente, un evento A puede iniciar antes B, pero terminar durante la vida de B.

Aspectos de diseño deseables:

1. No intrusivo: (R9) El campo “Qué” debe permitir almacenar información variable, organizada en estructuras de datos que no se conocen de antemano y que se ajustan a diversos módulos de procesamiento. Se espera que la LTM no requiera dependencias de módulos externos para poder funcionar y representar los datos. A la vez, no puede depender en que los otros módulos no cambien la representación de sus datos.
2. Eficiente: (R10) El sistema debe ser lo suficientemente eficiente para tolerar el manejo de una alta tasa de eventos, sin degradar el funcionamiento del robot. Es decir, todos los eventos deben ser procesados eventualmente, aún cuando el robot esté ocupando gran parte de sus recursos, y sin generar hambruna de CPU ni ancho de banda (de disco y red) al resto de los procesos.
3. Escalable: (R11) Los costos asociados al manejo de la información (agregar, eliminar, actualizar y buscar datos) en la memoria deben escalar bien, respecto a la cantidad de datos almacenados. La memoria debe mantener los costos acotados, dentro de un rango que no entorpezca su uso.

Procesos de consolidación y deterioro

En un esquema LTM, un episodio puede estar constituido de muchos eventos, pero no todos son igualmente relevantes. En su trabajo, Kelley [13] estudia 3 estrategias para la consolidación de recuerdos. La primera almacena todos los eventos ocurridos, pero tiene un costo de búsqueda lineal, respecto a los eventos almacenados; esta estrategia es impráctica a largo plazo. La segunda sólo almacena eventos interesantes y realiza una búsqueda entre los más recientes; esta estrategia es práctica, pero no permite abstracción del evento. La tercera estrategia se basa en un postprocesamiento de las memorias, de manera similar al sueño humano.

La estrategia propuesta por Kelley se basa en recordar todos los eventos, pero realizando un postprocesamiento de los datos una vez terminado el episodio. La ventaja es que no sólo permite recordar los eventos interesantes del episodio, sino que además permite reconocer pistas o estímulos previos que sirven para prevenir un evento indeseado o potenciar eventos interesantes. Además, permite almacenar eventos posteriores, que sirven para entender las consecuencias del evento de interés.

En la Figura 2.2 se muestra un episodio conformado de una secuencia de 9 eventos. Entre los marcadores 1-2 y 5-6 hay eventos considerados poco interesantes, mientras que los eventos entre 3-4 son interesantes. Kelley propone almacenar la secuencia completa de eventos, para descartar los que no son útiles en un postprocesamiento. A priori se deben quitar los eventos entre 1-2 y 5-6, sin embargo, el evento 2 se mantiene en la Ep-LTM como pista, y el evento 5 se mantiene para reforzar la consecuencia del episodio. Los resultados se pueden utilizar para aprendizaje reforzado, mientras que las pistas sirven para generalizar el episodio, en caso de que sean recurrentes.

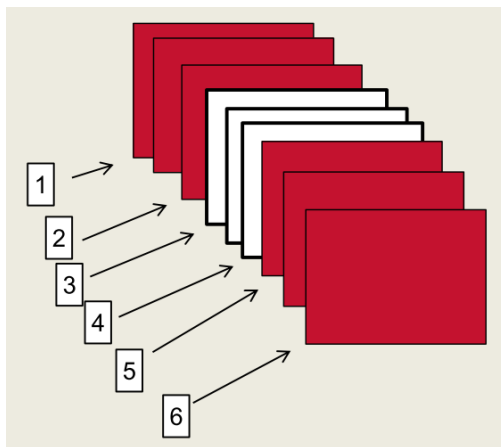


Figura 2.2: Ejemplo de una secuencia de eventos. Los cuadros coloreados y blancos indican eventos con poca o mucha relevancia, respectivamente. Los marcadores indican transiciones entre fases de poco y mucho interés. Obtenido de [13].

Además, este diseño permite que el sistema cambie su opinión sobre una evento, mediante aprendizaje reforzado. Si se repiten eventos, pero la consecuencia deja de ser la misma, entonces el sistema se acostumbra.

2.3.4. Memoria Emocional

La importancia de un evento se ve fuertemente influenciada por el estado emocional de una persona. Por lo tanto, la decisión de que almacenar o recordar depende de las emociones [14].

Su implementación requiere como mínimo de un mapeo entre estímulos percibidos por

el robot y las sensaciones emocionales que estos generan. Dood et al. [15] propone el uso de la teoría emocional de reacciones de Haikonen, que considera a una emoción como una combinación de estímulos básicos. Las sensaciones elementales son: bienestar, malestar, dolor, placer e interés.

Dood et al. proponen implementar las sensaciones a partir de distintos estímulos medidos en un robot:

- Actuador que se aproxima a sus límites de movimiento físico o de fuerza.
- Nivel de iluminación percibido.
- Nivel de ruido acústico percibido.
- Ausencia o presencia de humanos. Falta de interacción.
- Cumplimiento de objetivos.
- Cumplimiento de expectativas.

Sistemas más avanzados, incluso pueden considerar la generación de reacciones emocionales, basándose en las sensaciones derivadas anteriormente. En la Figura 2.3 se muestran las reacciones generadas según el modelo de Haikonen. Además, estas se podrían reflejar en la personalidad del robot, por ejemplo, mediante gestos, vocabulario o nivel de aceptación para realizar una acción. Kasap et al. [16] utilizan un sistema llamado *Emotion Engine*, para generar reacciones emocionales y simular cambios de personalidad de un robot, según las sensaciones percibidas.

Sensación Elemental	Reacción
Bueno: gusto, aroma	Aceptación, Acercar
Malo: gusto, aroma	Rechazo, Alejar
Dolor: autoinfligido	Alejar, Desistir
Dolor: agente externo	Agresión
Dolor: sobre esfuerzo	Sumisión
Placer	Mantener, Acercar
Acierto	Mantener atención
Desacierto	Migrar atención
Novedad	Enfocar atención

Figura 2.3: Sensaciones elementales y sus reacciones correspondientes, según el modelo de Haikonen. Obtenido de [15].

Para su uso efectivo dentro de un esquema LTM, se espera que las sensaciones reportadas incluyan un nivel de intensidad. Según el nivel percibido en cada episodio, es posible clasificarlos entre eventos muy o poco relevantes. Los más relevantes tendrán mayor probabilidad de ser recuperados al recordar. Deutsch et al. [14] consideran que la intensidad de las

sensaciones es importante, pues permite evitar costos de búsqueda lineales dentro de todos los episodios almacenados. Por otro lado, Dood et al. proponen curvas de decaimiento para la importancia de los episodios, que permiten simular la pérdida de interés en los eventos.

2.3.5. Otros enfoques

A continuación se presentan algunos estudios relacionados con aspectos de una memoria Ep-LTM que escapan de los requerimientos para este proyecto o que simplemente no son basados en la taxonomía de la memoria humana. Estos trabajos sólo se presentan a modo de completitud, pues no permiten resolver el objetivo de este proyecto, sino que sólo comprender otros enfoques y acercamientos a la solución.

El sistema propuesto por Ho et al. [9] busca modelar la memoria de forma suficientemente general, como para permitir el traspaso de los recuerdos de un robot a otro, independientemente de que el hardware sea distinto; El costo de esto, es que se reduce la personalización de cada robot. Ho et al. además aplican la teoría *Roboética*, sugerida por Veruggio y Operto [17], de donde derivan restricciones de diseño, relativas al manejo de información privada de los usuarios.

En [18], Kim et al. plantean el uso de Deep Learning para modelar la memoria episódica y la planificación de acciones de manera holística. En su implementación, los procesos de codificación, almacenado y recuperación de episodios son manejados como uno solo. Los procesos de decaimiento y relevancia son abstraídos, para ser manejados automáticamente por la red.

Thorsten et al. [19] proponen una memoria LTM para el robot BIRON. En su trabajo, se abstraen de la clasificación entre memorias Ep-LTM y S-LTM, pues todos los datos de largo plazo almacenados por el robot son considerados LTM. La memoria almacena sólo datos de alto nivel, obtenidos tras el procesamiento de streams de datos básicos, como cámaras, micrófonos o actuadores. Los datos almacenados corresponden a un historial de percepciones y acciones de alto nivel realizadas, como: detecciones de objetos, interacciones verbales o la descripción de movimientos realizados. A pesar de su simplicidad, esta arquitectura centralizada permite reducir las dependencias entre sí de cada componente y reducir el ancho de banda utilizado para retransmitir la información entre procesos.

Capítulo 3

Aspectos Técnicos

La implementación de un sistema de memoria robótica asume que muchos sistemas y capacidades de un robot están disponibles. Entonces, se requiere del uso de variados frameworks y librerías que permiten comunicarse con el robot y acceder a los datos de interés que se desea recordar. En este capítulo se presenta el software de interés para la implementación de la memoria y se detalla la plataforma objetivo.

3.1. ROS

Los sistemas robóticos actuales son cada vez más complejos. Deben lidiar con muchos componentes tanto de hardware como de software y su interacción, de una forma eficaz y que no entorpezca el desarrollo. Muchas tareas de control requieren altas frecuencias de funcionamiento, así como la sincronización y comunicación entre los diversos módulos. Por lo tanto, el cómo se unen los subsistemas en una aplicación robótica es una tarea difícil.

ROS [2], acrónimo para Robot Operating System, es un proyecto que funciona como middleware para aplicaciones robóticas, y permite resolver el problema de la comunicación entre procesos. Es una colección de herramientas, librerías y convenciones que buscan simplificar la tarea de crear comportamientos robóticos complejos y robustos, sin importar la plataforma robótica.

Fue originalmente creado por la organización WillowGarage en el 2008, y mantenido actualmente por la Open Source Robotics Foundation (OSRF). Existe un ecosistema ROS, mantenido por la comunidad, y con cientos de módulos de software con soluciones a problemas específicos, los que pueden interconectarse para construir comportamientos más complejos. Por lo anterior, su uso se ha convertido en una práctica mundial, siendo adoptado incluso en soluciones industriales.

3.2. UChile ROS Framework

UChile ROS Framework (URF) hace referencia al sistema de software desarrollado en el laboratorio de robótica del Departamento de Ingeniería Eléctrica de la Universidad de Chile, para sus robots de servicio. El sistema cuenta con 10 años de desarrollo y está orientado a cumplir los requisitos de la competencia Robocup en su categoría @Home.

URF está construido sobre ROS y en una estructura de 4 capas. La primera capa contiene todas las dependencias del sistema, ya sean de ROS o no; Es la única capa que contiene sólo código externo. Sobre ella, se monta una capa ROS de bajo nivel, con herramientas y librerías comunes, sumado a los drivers necesarios para manejar cada robot (módulos de *Hardware* en la Figura 3.1). La capa intermedia alberga capacidades robóticas avanzadas, relacionadas a percepción robótica, manipulación de objetos, navegación autónoma e interacción Humano-Robot (sección *ROS modules* del diagrama). Finalmente, existe una capa *de alto nivel* desarrollada en Python, con interfaces para el uso de las capacidades de menor nivel, utilizada para la elaboración de maquinas de estado y comportamientos robóticos complejos (cuadros *ROS-python interface* y *State Machine* del diagrama).

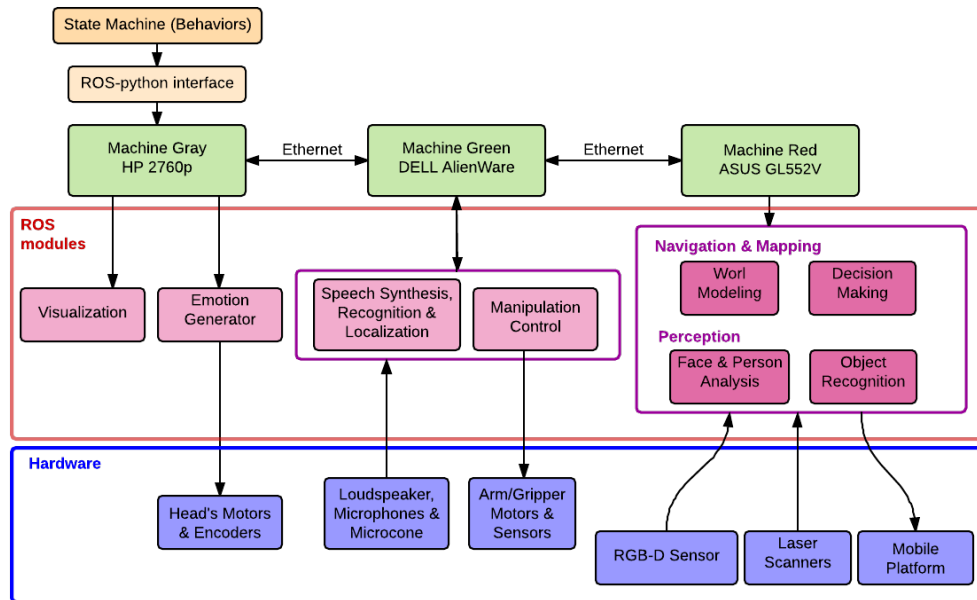


Figura 3.1: Diagrama de UChile ROS Framework utilizado en el robot Bender.

Todos los módulos de URF son de código libre, a excepción de los algoritmos relacionados con percepción y la interfaz de alto nivel. El código se almacena públicamente en la organización *uchile-robotics* en GitHub¹.

¹Organización *uchile-robotics* y URF en GitHub: <https://github.com/uchile-robotics>

Manejo de información en URF

Si un robot implementa URF, entonces es posible acceder a la información compartida por sus procesos. Cualquier módulo ROS en el sistema tiene acceso a los datos extraídos desde sensores y luego generados en post-procesamientos, junto al acceso para controlar el hardware.

Existen algunas formas de memoria implementadas en URF, comparables a los conceptos definidos para la memoria humana. También se pueden dividir en de corto y largo plazo:

Como STM, se puede definir como memoria de trabajo a todo el flujo de información presente durante la ejecución del robot. Lo que incluye datos sensados, procesamiento y acciones realizadas. Generalmente tales datos no son almacenados para posteriores ejecuciones.

A manera de LTM, se puede encontrar una memoria procedural, relacionada con todo el conocimiento almacenado que posee el robot para cumplir ciertas tareas. Caen en esta categoría: modelos para percepción robótica, modelos para reconocimiento de voz y patrones, bases de datos de movimientos precalculados para manipular objetos y acciones predefinidas que se utilizan para controlar el robot.

También se pueden encontrar especializaciones de memoria LTM semántica. Ejemplos de esto son: El mapa que se conoce del entorno, junto a los lugares y objetos anotados en él. Diccionarios con información anotada sobre entidades y sus características, cómo personas y objetos. Bases de datos con imágenes anotadas para el reconocimiento de objetos y personas.

Sin embargo, en URF no existen formas de memoria emocional ni episódica de largo plazo. Luego, toda interacción realizada por los robots está limitada a la información obtenida desde el inicio al término de cada rutina.

3.3. KnowRob

KnowRob es un sistema de procesamiento de conocimiento. Combina métodos para representar conocimiento y para razonar a partir de él, junto a técnicas para adquirir conocimiento y almacenarlo físicamente. Permite integrar información proveniente de distintas fuentes [20, 21].

Está implementado en Java y Prolog, y provee una interfaz ROS. Almacena los datos en archivos OWL (Web Ontology Language) y en una base de datos NoSQL, MongoDB. Por su diseño, permite el manejo e inferencia de memoria semántica y procedural.

3.4. Plataforma objetivo: Robot Bender

Bender es un robot humanoide creado el año 2007 en el laboratorio de robótica del Departamento de Ingeniería Eléctrica de la Universidad de Chile. El equipo UChile Homebreakers

es el encargado de su desarrollo y su objetivo es ser un mayordomo para el hogar, funcionando de manera autónoma para apoyar en tales labores [1].

En cuanto a actuadores, el robot cuenta con 2 brazos antropomórficos de 6 grados de libertad cada uno, una base móvil diferencial Pioneer 3-AT, un cuello que permite rotaciones en dos ejes cartesianos; pudiendo imitar gestos de asentimiento y negación, y finalmente, una cabeza que puede mostrar expresiones faciales mediante movimientos de su boca, orejas, cejas y cambios de colores alrededor de los ojos.

El robot cuenta con los siguientes sensores: un laser Hokuyo UTM-30LX, un laser Hokuyo URG-04LX-UG01, un micrófono M-Audio Producer USB y una cámara de profundidad ASUS Xtion Pro.

El software de Bender está basado en el framework URF. Su arquitectura de software utiliza ROS para el manejo de componentes de bajo y medio nivel. La capa de alto nivel, escrita en python, se abstrae de ROS y permite la creación de comportamientos complejos mediante máquinas de estado. Todos los módulos que interactúan con sensores y actuadores están implementados en ROS.



Figura 3.2: Robot Bender en competencia RoboCup 2015.

Capítulo 4

Metodología

En este capítulo se presenta la metodología a utilizar para el desarrollo del trabajo de título. En primer lugar se presentan las alternativas de solución evaluadas para el proyecto, haciendo énfasis en la factibilidad de la propuesta. Para luego, describir la planificación de tareas a realizar durante el segundo periodo del proceso, sustentadas en las soluciones propuestas. Finalmente, se presenta una carta gantt con los periodos de trabajo estimados.

4.1. Propuesta de Solución

La primera parte consiste en definir un conjunto de consultas que un robot doméstico debe responder a partir de la memoria. Esto estará basado en los requerimientos mostrados en la Sección 2.1. Particularmente, el foco del trabajo, será generar una memoria LTM adecuada a las interacciones que realizaría el robot Bender. A partir de las consultas se definirán los modelos semánticos a considerar y sus características.

El diseño de la memoria LTM estará basado en los 11 requerimientos de diseño mostrados en la sección 2.3.3. Además se basará en la taxonomía de la memoria humana, de manera similar a los trabajos desarrollados por Vijayakumar [6] y Sánchez et al. [7].

El algoritmo a utilizar para la consolidación de STM en LTM no está claro aún y debe ser estudiado. Esta es la parte más relevante del trabajo, donde se tomarán las decisiones de diseño definitivas. No existe un consenso al respecto en la literatura, pero un primer acercamiento será basado en la propuesta de Sánchez et al. [7], para la definición de episodios, sumado al trabajo de Kelley [13] para la selección de qué eventos almacenar y cómo procesarlos.

Una vez definido el proceso de consolidación, queda definir una arquitectura de software. Se propone basar el desarrollo en el software KnowRob, pues fue diseñado para manejar memoria semántica y procedural, además de proveer funcionalidades para realizar inferencias. Por su estructura, se cree que la Ep-LTM puede ser almacenada como si fuese memoria semántica. Entonces, el desafío se centra en poder agregar el soporte para memoria episódica al

sistema. KnowRob ha sido descargado y se ha comprobado el funcionamiento de sus módulos principales.

En cuanto a la integración de KnowRob y la Ep-LTM en URF, se cree que se deberá agregar un nuevo módulo al sistema. De acuerdo a la Figura 3.1, se debería agregar la Ep-LTM como un componente en la *capa de alto nivel*, pues es donde se manejan las máquinas de estado del robot y el único lugar donde se podrían abstraer los conceptos de evento y episodio. Las capas de menor nivel sólo se encargan de proveer funcionalidades para construir los comportamientos ejecutados por el robot. Se debe diseñar la solución final, pero a priori, la Ep-LTM debe tener acceso a todos los componentes de la *capa de alto nivel*. Además, se debe considerar que una futura implementación de memoria emocional debería interactuar directamente con el hardware del robot.

Finalmente, se debe implementar un servicio que escuche constantemente los eventos ocurridos en el sistema, los analice y almacene recuerdos. Luego se debe implementar una demostración del uso de los datos recopilados.

No se trabajará en la inferencia de información, sino que sólo se aprovechará de que KnowRob provee herramientas para ello, lo que asegura que en un futuro se podrán agregar inferencias al sistema. La memoria emocional sólo será considerada para la etapa de diseño, pero su implementación no es requerida.

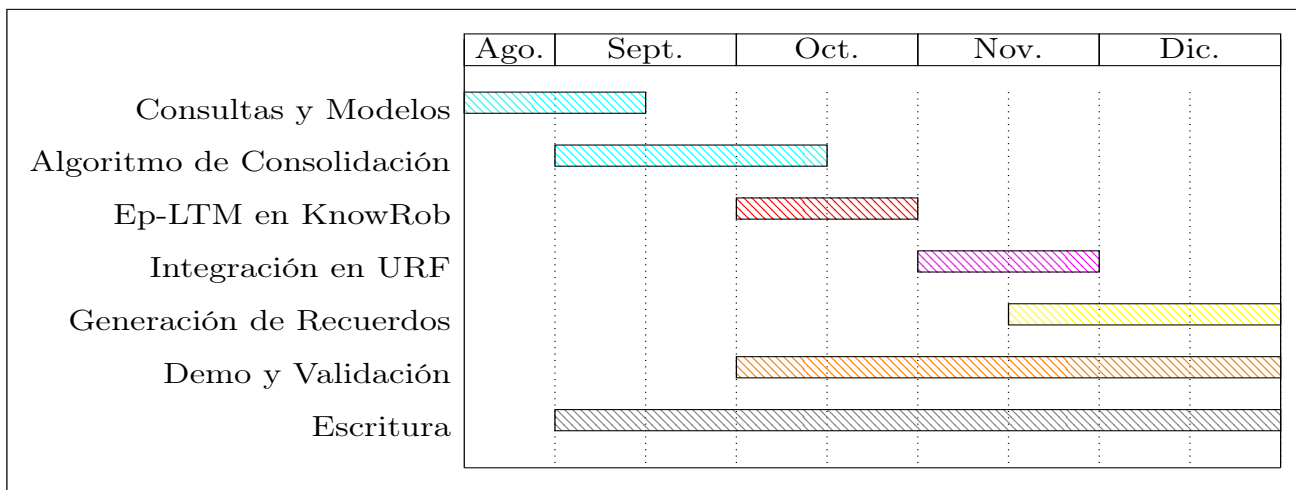
La demostración será definida como un conjunto de validaciones que permitan corroborar el funcionamiento del sistema. Por lo tanto, esta será implementada incrementalmente, para validar cada desarrollo del trabajo. Las validaciones más relevantes tienen que ver con la capacidad del sistema para responder las consultas episódicas seleccionadas para robots domésticos.

4.2. Planificación

Durante el trabajo de título se propone seguir una estrategia incremental de desarrollo de software, que considere cada uno de los objetivos específicos. La demostración es de carácter transversal al proyecto, pues se utilizará como medio de validación de cada parte del trabajo. Considerando la solución propuesta, se tienen las siguientes etapas del proyecto:

1. Definir consultas a responder por un robot doméstico. Seleccionar entidades a ser modeladas, acordes a las consultas.
2. Diseñar el algoritmo de consolidación de STM en LTM.
3. Implementación de componente Ep-LTM para KnowRob.
4. Integración de KnowRob en Bender
5. Servicio para recopilación continua de recuerdos.

Los tiempos estimados de trabajo para cada una de las tareas anteriores se presentan en la siguiente carta Gantt. La resolución de los tiempos es a periodos de 15 días.



Bibliografía

- [1] UChile Robotics. [Online]. Available: <http://robotica-uchile.amtc.cl/>
- [2] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Mg, “ROS: an open-source Robot Operating System,” in *Icra*, vol. 3, 2009, p. 5.
- [3] SoftBank Robotics. [Online]. Available: <http://www.softbank.jp/en/robot/>
- [4] L. van Beek, K. Chen, D. Holz, L. Sanchez, M. Matamoros, H. Nagano, C. Rascon, J. R. de Souza, M. Rudinac, and S. Wachsmuth, “RoboCup@Home 2017: Rules and Regulations,” [\url{http://www.robocupathome.org/rules/2017_rulebook.pdf}](http://www.robocupathome.org/rules/2017_rulebook.pdf), 2017.
- [5] H. Eichenbaum, “Memory,” *Scholarpedia*, vol. 3, no. 3, p. 1747, 2008.
- [6] S. Vijayakumar, “Long-Term Memory in Cognitive Robots,” Ph.D. dissertation, Universitaet des Saarlandes, 2014.
- [7] M. L. Sánchez, M. Correa, L. Martínez, and J. Ruiz-Del-Solar, “An episodic long-term memory for robots: The bender case,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9513, pp. 264–275, 2015.
- [8] IFR, International Federation of Robotics. [Online]. Available: <https://ifr.org/>
- [9] W. C. Ho, M. Y. Lim, P. A. Vargas, S. Enz, K. Dautenhahn, and R. Aylett, “An Initial Memory Model for Virtual and Robot Companions Supporting Migration and Long-term Interaction,” pp. 277–284, 2009.
- [10] R. Salgado, F. Bellas, P. Caamano, B. Santos-Diez, and R. J. Duro, “A procedural Long Term Memory for cognitive robotics,” *2012 IEEE Conference on Evolving and Adaptive Intelligent Systems*, pp. 57–62, 2012.
- [11] D. Stachowicz and G. J. M. Kruijff, “Episodic-like memory for cognitive robots,” *IEEE Transactions on Autonomous Mental Development*, 2012.
- [12] S. Jockel and M. Weser, “Towards an episodic memory for cognitive robots,” *European Conference on Artificial Intelligence*, pp. 68–74, 2008.

- [13] T. D. Kelley, “Robotic Dreams: A Computational Justification for the Post-Hoc Processing of Episodic Memories,” *International Journal of Machine Consciousness*, vol. 06, no. 02, pp. 109–123, 2014.
- [14] T. Deutsch, A. Gruber, R. Lang, and R. Velik, “Episodic Memory for Autonomous Agents,” pp. 2–7, 2008.
- [15] W. Dodd and R. Gutierrez, “The role of episodic memory and emotion in a cognitive robot,” *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, vol. 2005, pp. 692–697, 2005.
- [16] Z. Kasap and N. Magnenat-Thalmann, “Towards episodic memory-based long-term affective interaction with a human-like robot,” in *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, 2010.
- [17] G. Veruggio and F. Operto, “Roboethics: A Bottom-Up Interdisciplinary Discourse in the Field of Applied Ethics in Robotics,” *International Review of Information Ethics*, vol. 6, pp. 2–8, 2006.
- [18] M.-J. Kim, S. H. Baek, S. H. Cho, and J. H. Kim, “Approach to integrate episodic memory into cogency-based behavior planner for robots,” in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, oct 2016, pp. 4188–4193.
- [19] T. P. Spexard, F. H. K. Siepmann, and G. Sagerer, “Memory-based Software Integration for Development in Autonomous Robotics,” *Intelligent Autonomous Systems 10*, pp. 49–53, 2008.
- [20] M. Tenorth and M. Beetz, “KnowRob: A knowledge processing infrastructure for cognition-enabled robots,” *The International Journal of Robotics Research*, vol. 32, no. 5, pp. 566–590, apr 2013.
- [21] —, “KNOWROB - knowledge processing for autonomous personal robots,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, oct 2009, pp. 4261–4266.