ESPECIALIZACIÓN EN INGENIERÍA EN SISTEMAS DE INFORMACIÓN

TRABAJO FINAL INTEGRADOR

"Un Análisis comparativo de arquitecturas de Inteligencia Artificial General para el desarrollo de Sistemas Multiagentes."

Autor: Ing. Darío Alberto Lamy

Directora: Dra. Milagros Gutierrez

Universidad Tecnológica Nacional – Facultad Regional Santa Fe

Año 2022

1 - Introducción

Para poder entender el contexto a partir del cual se plantean los objetivos del trabajo, se presenta una introducción general de dos conceptos fundamentales: Inteligencia Artificial (AI) e Inteligencia Artificial General (AGI).

1.1 - Inteligencia Artificial

Inteligencia Artificial General es la inteligencia de las máquinas que le permiten comprender, aprender y ejecutar actividades intelectuales como los humanos.

La AGI intenta emular la mente humana y su comportamiento para resolver cualquier tipo de problemas complejos. Se enfrenta al entrenamiento y aprendizaje del comportamiento humano como así entendimiento de los principales aspectos de la conciencia. Con esta base tan fuerte, esta inteligencia podría planear y aprender habilidades cognitivas, hacer juicios de valor, manejar situaciones inciertas, integrar conocimiento previo en la toma de decisiones para mejorar su precisión entre tantas otras. [1]

Luego puede expandirse esta definición y analizarse desde diferentes enfoques. Algunos empíricos, donde se involucran el pensamiento y comportamiento humano, hasta otros racionalistas, donde se priorizan el pensar y actuar racionalmente. [2]

Dentro de los enfoques empíricos podemos destacar dos, el actuar como humano y el pensar como humano.

El actuar como humano comprende principalmente el pasar satisfactoriamente el test de Turing, propuesto por Alan Turing en los 50. Esta inteligencia requeriría de procesamiento del lenguaje natural, representación del conocimiento, razonamiento automático y aprendizaje maquinal.

Luego, el pensar como humano apunta al modelo cognitivo, donde se pretende que esta inteligencia artificial piense de la forma que pensamos los humanos. Aquí la principal dificultad radica en encontrar de qué forma pensamos, para poder así imitarla.

Dentro los enfoques racionalistas también podemos destacar dos, el actuar racionalmente y el pensar racionalmente.

El pensar racionalmente se basa principalmente en el uso de la lógica para obtener resultados a partir de precondiciones.

Por último, el actuar racionalmente introduce el modelo de agente, donde existe un agente inteligente que percibe, procesa y actúa de acuerdo a los modelos de él mismo y del entorno que ha generado para alcanzar sus objetivos, pero todo esto de forma autónoma. [2]

Como puede verse no existe una única definición de inteligencia artificial y dependiendo del enfoque que se quiera tomar se tendrán distintas apreciaciones de la misma.

Desde el punto de vista teórico, cada uno de estos enfoques fundamentales se puede abordar de un modo genérico, no obstante, de querer llevarlos a una aplicación práctica se debe acotar su campo de acción de acuerdo a un dominio; esto impacta desde la representación del conocimiento necesaria hasta los tipos de lenguajes y percepciones utilizadas.

Es desde aquí donde aparece una división clara entre tipos de Inteligencia artificial, una general y multipropósito, capaz de resolver cualquier problema que se le presente con un nivel de pensamiento o razonamiento similar al humano y otra enfocada a problemas particulares, donde con un dominio acotado y ambientes controlados pueden generar resultados incluso mejores a los cuales los humanos puedan alcanzar. Esta inteligencia artificial multipropósito se la conoce como Inteligencia Artificial General o de su nombre en inglés AGI (Artificial General Intelligence).

1.2 - Inteligencia Artificial General vs Inteligencia Artificial Tradicional

La Inteligencia Artificial tradicional, comprende por lo general, algoritmos avanzados que siguen funciones matemáticas o reglas que son capaces de realizar procesos de alta complejidad de igual o mejor manera que los humanos. Dentro de estos procesos se encuentran: reconocimiento de imágenes, reconocimiento del habla, traducciones de un idioma a otro, sistemas recomendadores para la toma de decisiones, conducción de automóviles, planeamiento de logísticas, juegos, reconocimiento de patrones, etc. [3]

Dentro de estas aplicaciones tienen un propósito específico y fueron o son creadas y entrenadas para tal propósito. De esta manera pueden verse como instanciaciones específicas de la inteligencia artificial.

Por otro lado, se llama Strong AI (Inteligencia Artificial Fuerte) o AGI (Artificial General Intelligence) a la inteligencia artificial en la cual computadoras o máquinas pueden obtener suficientes habilidades de pensamiento y/o conciencia para encarar y resolver problemas multipropósito [4]. También puede ser vista como sistemas capaces de realizar exitosamente tareas en diferentes dominios resolviendo problemas [5], o sistemas con la capacidad de aprender diferentes habilidades y aprender a utilizarlas en diferentes circunstancias y ambientes [6].

Existen muchas otras definiciones o visiones acerca de la AGI pero en esencia todas apuntan a la importancia de poder desempeñarse de manera exitosa resolviendo problemas o situaciones en diferentes dominios y ambientes.

Este enfoque radicalmente diferencial entre la AI tradicional y la AGI hace que, tanto las teorías aplicables como las herramientas a utilizar sean muy distintas y con requerimientos muy dispares.

1.3 - Fundamentación

Introducidas las definiciones anteriores de inteligencia artificial tradicional (AI) e Inteligencia Artificial General, y para acercarse a los objetivos de este trabajo de investigación, se introducen las arquitecturas de AGI, identificando la importancia de contar con un mecanismo que permita su comparación con propósitos particulares, y la identificación de las herramientas que se utilizan para su utilización y desarrollo.

1.3.1- Arquitecturas de AGI

Una arquitectura de AGI contempla la definición de todas las partes que interactúan para cumplir el objetivo de tener la capacidad de una inteligencia multipropósito. Se han creado distintos modelos de arquitecturas que permiten en última instancia, una vez implementados, dar solución y generar esta inteligencia artificial general. Ejemplos de ellas son: Lida Framework [7], Soar Cognitive Architecture [8] y OpenCog [9].

Cada uno de estos modelos de arquitecturas de AGI define todos los aspectos necesarios para cumplir sus objetivos, desde la representación del conocimiento, hasta el aprendizaje y toma de decisiones, entre otras cosas.

1.3.2 - Comparativas existentes entre arquitecturas

Si bien en BICA society [10], se presenta un análisis comparativo muy completo de arquitecturas de AGI, en donde se comparan un gran número de arquitecturas utilizando diferentes criterios comunes, no se ha actualizado desde 2012. Además, los criterios de comparación son genéricos y no aplicados a ningún dominio específico, por tratarse naturalmente de AGI. No obstante, para este trabajo se quiere realizar un análisis comparativo enfocado en el desarrollo de sistemas multiagentes, por lo que se requerirá que este análisis sea más crítico, dedicado y selectivo a los aspectos de importancia en el desarrollo de sistemas multiagentes.

Para la elección de las arquitecturas a comparar, también se tomarán en cuenta el avance en frameworks desarrollados en base a la definición de las arquitecturas ya que se pretende tener como trabajo a futuro implementaciones con la arquitectura elegida, una vez hecha la comparación.

1.4 - Objetivos

1.4.1 - Objetivo General

El objetivo general de este trabajo es realizar un análisis comparativo de arquitecturas de AGI que den soporte al proceso de selección de las mismas para el desarrollo de un sistema multi-agentes.

1.4.2 - Objetivos Específicos

- Especificar las ventajas de la AGI respecto de la AI tradicional
- Identificar y describir los principales campos de acción e inconvenientes a los que se enfrenta AGI.
- Establecer criterios de análisis y definir indicadores a partir de los cuales se realizará la comparación entre arquitecturas AGI
- Diseñar una herramienta de análisis que sirva de base en la selección de arquitecturas AGI para el desarrollo de sistemas multi-agentes.

2 - Campos de acción e inconvenientes de la AGI

2.1 - Campos de acción

Las aplicaciones de un sistema con inteligencia artificial general al nivel de la inteligencia humana son tan incontables como las aplicaciones que nuestra inteligencia pueda tener. Esto contempla por ejemplo realizar acciones correctivas en plantas nucleares, colapsos de mercados financieros, mejorar la producción de alimento, ser un tutor para los niños, cuidar la casa cuando los dueños no se encuentren, realizar las actividades cotidianas de limpieza, etc. La variedad es en definitiva proporcional a la imaginación, ya que un sistema con este nivel de inteligencia tendría la capacidad de aprender a desenvolverse y realizar acciones en cualquier ambiente con el tiempo adecuado de aprendizaje y las herramientas correctas, al igual que los humanos.

Si se analizan las actividades humanas capaces de realizar y entender, la AGI tendría la capacidad de auto mejorarse, tanto en diseño como en estructura, lo cual la llevaría a un ciclo de mejoras que sería difícil incluso de conceptualizar.

Esto no quiere decir que las aplicaciones de la AGI sean radicalmente distintas o incompatibles a las de la AI actual, las cuales se crean con un propósito específico, sino que son en cierta manera complementarias. Para algunas aplicaciones, quizás el nivel de complejidad de entendimiento requerido, no involucre consciencia o sentimientos acerca de lo que está haciendo, y solo lo debe realizar con la precisión y capacidad necesaria. En cambio habrá otras que requerirán un entendimiento del contexto y experiencias previas para las cuales no fue entrenada específicamente y deberá encontrar la manera de hacerlo, similar a la capacidad humana. [15]

2.2 - Dificultad de medir el progreso parcial y de realizar pruebas

Existe una dificultad grande para determinar qué tan cercana al objetivo de imitar la inteligencia humana está una determinada arquitectura de AGI, dada la fuerte integración que existen entre sus partes. Por ejemplo implementando el 75% de la arquitectura no dará como resultado necesariamente una inteligencia con el 75% de la capacidad humana, y en realidad probablemente ni siquiera funcione.

Luego, también incluso teniendo completamente desarrollada e implementada la arquitectura, no existen formas sencillas y obvias de evaluar que se ha alcanzado algún nivel de inteligencia. Existen tests conocidos para evaluar este tipo de inteligencia artificial, como lo son el test de Turing [11] y el “coffee test” de Wozniak [12], pero siempre se corre el riesgo de estar persiguiendo el objetivo de pasar el test y de esta manera, hacer implementaciones que permitan, en cierta medida, hacer un “truco” para pasarlo y no pasar el test realmente por haber alcanzado el nivel de inteligencia deseado [13].

Se puede concluir que la principal dificultad en el desarrollo de las arquitecturas de AGI es que tener una implementación parcial o incompleta de la misma probablemente no funcione en absoluto, como no lo haría una porción aislada de nuestro cerebro y realizar cualquier tipo de test sobre la misma tampoco daría resultados valiosos [14].

3 - Marco de comparativa entre arquitecturas

3.1 - Sistemas multi-agentes

Antes de explicar de qué se trata un sistema multi-agentes se debe introducir el concepto de *agente*. Un *agente* es un sistema capacitado para actuar de manera autónoma en beneficio de su usuario o dueño. Es decir, un *agente* determina por sí mismo qué necesita hacer para satisfacer sus objetivos en vez de que se le tenga que ordenar explícitamente qué realizar en un momento determinado [27].

Otra definición del mismo es que un *agente* es cualquier cosa que pueda percibir en su ambiente a través de sensores y actuar en el mismo a través de actuadores [28].

Más específicamente, estas son características que poseen los agentes:

Autonomía: un agente puede operar sin la directa intervención de humanos u otros agentes.

Habilidad Social: un agente es capaz de interactuar con otros agentes a través de un lenguaje de comunicación de agentes.

Racionalidad: un agente puede razonar acerca de datos percibidos a fin de calcular una solución óptima.

Reactividad: un agente es capaz de percibir estímulos del entorno y estos estímulos guían las acciones del agente en su entorno.

Pro-actividad: un agente no es sólo una entidad que reacciona a estímulos, sino también tiene un carácter emprendedor y puede actuar guiado por sus propios objetivos.

Adaptabilidad: esta característica está relacionada con el aprendizaje que un agente puede lograr y con su capacidad para cambiar su propio comportamiento basado en este aprendizaje.

Movilidad: es la capacidad de un agente para moverse a través de una red.

Veracidad: un agente no puede comunicar información falsa de manera deliberada.

Benevolencia: un agente está dispuesto a ayudar a otros agentes si esto no está en contra de sus propios objetivos [29].

Un sistema multi-agentes consiste en un número de agentes que interactúan los unos con los otros, típicamente comunicándose a través del intercambio de mensajes. Para interactuar exitosamente estos agentes deberán tener la habilidad de cooperar, coordinar y negociar entre ellos [27].

Otra definición es que un sistema multi-agentes es una comunidad de entidades autónomas donde cada una percibe, decide y actúa por sí mismas de acuerdo a su propio interés pero que también podría cooperar con otras para conseguir objetivos y metas comunes [31].

Hay aspectos fundamentales que distinguen los sistemas multi-agentes de los sistemas de mono-agente y pueden ser categorizados en diferentes dimensiones, que son:

Diseño de agente: El diseño entre diferentes agentes puede variar tanto en hardware (por ejemplo, en caso de robots diferentes plataformas mecánicas) como en software (por ejemplo, agentes ejecutándose en diferentes sistemas operativos). En estos casos se llamará a los agentes heterogéneos en contraste de los agentes homogéneos donde son diseñados de manera idéntica y tienen a priori las mismas capacidades. Pero no necesariamente esa será la única distinción posible, ya que agentes basados en el mismo hardware y software que implementan diferentes comportamientos también serán heterogéneos. Este problema de heterogeneidad puede afectar todos los aspectos funcionales de un agente desde la percepción hasta la toma de decisiones, mientras que en sistemas basados en un solo agente este problema no existe.

Ambiente: Los agentes tienen que actuar en ambientes que pueden ser estáticos o dinámicos. La mayoría de las técnicas existentes de AI para mono-agentes han sido desarrolladas para ambientes estáticos, ya que son más fáciles de manejar y permiten un trato matemático más riguroso. En un sistema multi-agentes, la mera presencia de múltiples agentes hace al ambiente parecer dinámico desde el punto de vista de cada agente.

Percepción: La información colectiva que llega a los sensores de los agentes en un sistema multi-agentes es típicamente distribuida. Los agentes pueden observar datos que difieren espacialmente, temporalmente e incluso semánticamente. Esto hace automáticamente al ambiente parcialmente observable para cada agente, lo que tienen múltiples consecuencias en la toma de decisiones de los agentes.

Control: Contrariamente a sistemas mono-agentes, el control en sistemas multi-agentes es típicamente distribuido. Esto significa que no existe un proceso central que recolecta información de cada agente y con ello decida qué acción debe tomar cada uno. La toma de decisiones de cada agente recae en ellos mismos. En un sistema multi-agentes cooperativo o de equipo, la toma de decisiones distribuida resulta en cálculos asincrónicos y ciertas aceleraciones, pero tiene también la contraparte de tener que desarrollar mecanismos de coordinación adicionales. Coordinación que asegure que las decisiones individuales de los agentes resultan en conjunto en una buena decisión para el grupo.

Conocimiento: En sistemas mono-agentes típicamente se asume que los agentes conocen sus propias acciones, pero no necesariamente como el ambiente es afectado por las mismas. En un sistema multi-agentes, los niveles de conocimiento de cada agente acerca del mundo pueden diferir sustancialmente. En general, en un sistema multi-agentes cada agente debe considerar también el conocimiento de los demás agentes en su toma de decisiones. Un concepto crucial aquí es el conocimiento común, de acuerdo al cual cada agente conoce un hecho, cada agente conoce que cada agente conoce ese hecho y así sucesivamente.

Comunicación: Interacción es usualmente asociada con alguna forma de comunicación. Típicamente se ve la comunicación en un sistema multi-agentes como un proceso de dos vías, donde todos los agentes pueden potencialmente ser remitentes o receptores de mensajes. La comunicación puede ser usada en diversos casos, por ejemplo, para la coordinación dentro de

agentes cooperativos o para la negociación entre agentes egoístas. Además, la comunicación también plantea los problemas de qué protocolos de red usar para que la información intercambiada llegue de manera segura y oportuna, y qué idioma deben hablar los agentes para entenderse (especialmente si son heterogéneos) [30].

3.2 - Criterios de comparación

Para la realización de un sistema mono-agente se podrían utilizar cualquiera de las arquitecturas de AGI ya que en sí mismas todas describen las dinámicas necesarias de percepción razonamiento y acción de un único ente.

Al querer implementarse un sistema multi-agentes con agentes individuales con Inteligencia Artificial General deben contemplarse las diferencias mencionadas de los sistemas multi-agentes respecto a los mono-agentes para hacer una elección de aquella que se alinee mejor con las necesidades.

Por ello para detectar cuál sería la más adecuada se establecerán criterios de comparación, basados puntualmente en las diferencias explicadas.

Diseño de agente: Un potencial problema en los sistemas multi-agentes es la heterogeneidad en el hardware y software de cada agente. Este, no sería un problema para este caso ya que todos utilizarían la misma arquitectura y se desenvolverá en el mismo tipo de hardware. De todas formas, relacionado al diseño del agente, se evaluará la capacidad de generar múltiples instancias dentro de recursos razonables de hardware para potencialmente evaluar el comportamiento emergente de la interacción de los agentes. Se considerará como cantidad tentativa 1000 agentes dentro del sistema.

Ambiente: Dentro del ambiente en el cual deben desenvolverse los agentes se evaluará la complejidad del ambiente con la que puede lidiar la arquitectura, la dinámica del mismo y la capacidad de escalar.

Percepción: Dado que el ambiente es parcialmente observable, se evaluará la capacidad de ser interactuado por múltiples agentes al mismo tiempo. Con lo cual, el ambiente donde se desenvuelve debe poder ser percibido y modificado al mismo tiempo por “n” número de agentes.

Control: La toma de decisiones no recae en un único agente, sino que cada uno toma sus propias decisiones. No obstante, si se tomasen objetivos generales, se requerirá la capacidad de coordinar grupos o equipos para distribuir tareas y realizar las acciones designadas. Por ello se evaluará la capacidad de poder comunicar directamente un agente a otro la decisión de una acción particular en busca de un objetivo común.

Conocimiento: En los sistemas multi-agentes cada agente debe tener la capacidad de interpretar que puede haber otro de su tipo con sus mismas capacidades, por lo que se evaluará la capacidad de incorporar el concepto de asumir que existe otro ente similar del

cual puede valerse de conocimiento para realizar las acciones. Es decir, en caso de que no se posea la información necesaria para determinar qué acción tomar en un contexto determinado, evaluar la posibilidad de conseguir esta de un agente externo; y no solo eso, sino también evaluarla.

Comunicación: Dada que la interacción entre agentes es fundamental en sistemas multi-agentes se evaluará la capacidad de transmitir información de todo tipo, ya sea conocimiento, memoria o decisiones hacia otros agentes, a través del intercambio de mensajes.

Rúbrica de evaluación

Con el objetivo de normalizar la comparación entre las arquitecturas que se van a elegir para evaluar teniendo en cuenta los criterios antes mencionados se presenta la siguiente rúbrica de evaluación.

La escala contendrá cinco divisiones (1-5) siendo 5 la mejor calificación con los siguientes significados:

Excelente (5): La arquitectura se adapta y favorece notablemente el desarrollo de la misma respecto del criterio seleccionado.

Muy bueno (4): La arquitectura facilita desde su estructura el desarrollo de la misma respecto del criterio seleccionado.

Bueno (3): La arquitectura se adapta correctamente para el desarrollo sin ser una ventaja intrínseca su modelo respecto al criterio seleccionado.

Regular (2): Si bien existen dificultades para el desarrollo, estas pueden ser salvables teniendo en cuenta restricciones respecto al criterio seleccionado.

Malo (1): Existen dificultades precisas para el desarrollo de la arquitectura respecto al criterio seleccionado.

A continuación, se describen las arquitecturas elegidas para la evaluación.

4 - Arquitecturas de la AGI

4.1 - Preselección de arquitecturas

Existen muchas arquitecturas de AGI, cada una con su propio nivel madurez, herramientas y/o frameworks creados para su implementación, equipos de trabajo o de investigación que la soportan, características de sus entradas, salidas, aprendizaje, tipos de memoria, paradigmas, etc. Se puede ver una clasificación y desagregado de características en la tabla "*Comparative Table of Cognitive Architectures*" [10].

En el presente trabajo se seleccionaron aquellas arquitecturas usadas en la resolución de problemas, de software libre y que permita la implementación de sistemas multi-agentes.

Los criterios para esta selección no están solo relacionados directamente con las teorías en las que se basan las arquitecturas o restricciones para su aplicación en sistemas multi-agentes, sino que se tomaron criterios de madurez de sus herramientas utilizadas, el hecho de ser o no de software libre, de ser capaz de ser aplicadas en resoluciones de problemas, conocimientos personales acerca del lenguaje en donde se encuentran estas implementadas, potencial utilización en extensión a múltiples agentes, aprendizaje por retroalimentación, toma de decisiones, orientación a objetivos y múltiples tipos de memorias.

Dicho esto, a continuación, se listan los criterios que se utilizaron para la preseleccionar arquitecturas de la comparativa de la tabla.

- Existencia de un framework desarrollado para poder realizar la implementación.
- Framework desarrollado en lenguajes orientados a objetos como java o C++.
- Framework de software libre.
- Existencia actual de investigación y soporte sobre la misma.
- Potencial utilización en extensión a múltiples agentes.
- Aprendizaje por retroalimentación (*reinforcement learning*) a través del cual agentes puedan aprender de la experiencia de interacción entre ellos y con el ambiente.
- Toma de decisiones (*decision making*), clave para orientar sus acciones a los objetivos particulares y/o colectivos.
- Orientación a objetivos (*goal oriented*), para poder desarrollar planes de acciones para maximizar sus esperanzas de cumplimiento de los mismos.
- Múltiples tipos de memorias, para representar el ambiente donde se desenvuelven, los agentes mismos y los demás agentes en la forma más adecuada según la circunstancia.

Con esos criterios en mente, luego de analizar las arquitecturas listadas en la tabla, se preseleccionaron las siguientes arquitecturas:

- LIDA
- COGPRIME
- CLARION

En la sección siguiente se describen cada una de ellas.

4.2 - Descripción de arquitecturas

Para poder realizar luego una comparación entre las arquitecturas preseleccionadas se necesita primero contar con la descripción de alto nivel de los componentes, modelos en los cuales se encuentran basados, conceptos abarcados y cualidades de los frameworks actualmente implementados. A continuación, se describen cada una de las arquitecturas seleccionadas.

4.2.1 - LIDA

4.2.1.1 LIDA basada en el modelo GWT

LIDA implementa GWT (Global Workspace Theory), la cual es una teoría psicológica y neurobiológica de la conciencia, que se describe brevemente a continuación.

GWT intenta integrar una gran cantidad de evidencia en un marco conceptual único centrado en el papel de la conciencia en la cognición humana. Como muchas otras teorías, GWT postula que la cognición humana es implementada por una multitud de procesadores relativamente pequeños y de propósito especial, mayormente inconscientes.

Estos procesadores son comparativamente simples, y la comunicación entre ellos es relativamente rara. Además, esta comunicación se da a través de un estrecho ancho de banda. Luego, una coalición de estos procesadores es una colección que trabaja en conjunto para realizar una tarea específica.

Las coaliciones normalmente realizan acciones rutinarias, en la búsqueda de tareas sensoriales, motoras u otras tareas de resolución de problemas.

GWT sugiere que el cerebro admite una capacidad global de *espacio de trabajo (workspace)* que permite la integración y distribución de estos procesadores.

Una coalición de procesadores que obtiene acceso al espacio de trabajo global puede transmitir un mensaje a todos los procesadores inconscientes, a fin de reclutar nuevos componentes para unirse en la interpretación de una situación novedosa o para resolver el problema actual dado. En GWT, la conciencia o procesos conscientes le permiten al cerebro lidiar con situaciones novedosas o problemáticas que no pueden ser tratadas de manera eficiente, o en absoluto, mediante procesos inconscientes habituales. GWT también sugiere una respuesta a la paradoja de la capacidad cognitiva limitada asociada con la experiencia consciente, la memoria inmediata y los objetivos inmediatos. GWT sugiere que la ventaja compensatoria es la capacidad de movilizar muchos recursos inconscientes de una manera no rutinaria para abordar nuevos desafíos.

GWT ofrece una explicación para que la conciencia sea de naturaleza serial en lugar de paralela, como es común en el resto del sistema nervioso. Los mensajes transmitidos en paralelo tienden a sobrescribirse unos a otros, lo que dificulta su comprensión. De manera similar, explica la capacidad limitada de la conciencia en oposición a la enorme capacidad típica de la memoria a largo plazo y otras partes del sistema nervioso.

LIDA es un modelo de prueba de concepto para GWT. Casi todas las tareas en este modelo se realizan mediante *codelets*, que representan a los procesadores en GWT. Los *codelets* son pequeños fragmentos de código, cada uno de ellos ejecutado de forma independiente.

El modelo LIDA también implementa la serialización de la conciencia en GWT con el ciclo cognitivo. Además, el modelo LIDA aborda una vasta extensión de procesos cognitivos que incluyen la percepción, varios sistemas de memoria, selección de acciones, mecanismos de aprendizaje evolutivos, sentimientos y emociones, deliberación, acción voluntaria, resolución de problemas no rutinarios y automatización.

4.2.1.2 Arquitectura de LIDA

La arquitectura LIDA es en parte *simbólica* y en parte conexionista (utilizando redes neuronales artificiales). Los mecanismos utilizados en la implementación de los diversos módulos se han inspirado en una serie de técnicas diferentes de "nueva IA".

Principales mecanismos de LIDA

1. Memoria asociativa perceptiva

   LIDA percibe exógena y endógenamente con los sistemas de símbolos perceptivos de Barsalou (1999). La base de conocimiento perceptual de este agente, llamada memoria asociativa perceptiva, toma la forma de una red semántica con activación llamada *slipnet*. Los nodos de la *slipnet* constituyen los símbolos perceptivos del agente, que representan a individuos, categorías, relaciones, etc.

2. Espacio de trabajo (*workspace*)

   El espacio de trabajo de LIDA es análogo a los *buffers* preconscientes de la memoria de trabajo humana. Los *codelets* de percepción escriben en el espacio de trabajo como lo hacen otros *codelets* más internos. Los *codelets* de atención miran lo que está escrito en el área de trabajo para poder reaccionar. Los elementos en el espacio de trabajo se deterioran con el tiempo y pueden sobrescribirse. En otras palabras el espacio de trabajo funciona como una especie de memoria global volátil donde todos los inputs obtenidos de los sensores y procesamientos iniciales son almacenados para luego ser leídos y tratados por los *codelets*.

3. Memoria episódica

   La memoria episódica en la arquitectura LIDA se compone de una memoria declarativa para el almacenamiento a largo plazo de información autobiográfica y semántica, así como una memoria episódica transitoria a corto plazo similar a la memoria episódica sensorial-perceptual de Conway (2001) con una tasa de retención medida en horas.

4. Memoria de procedimiento

   La memoria de procedimiento en LIDA es una forma modificada y simplificada del mecanismo de esquema de Drescher (1991), *scheme-net*. Al igual que *slipnet* de la memoria asociativa perceptiva, la *scheme-net* es un grafo dirigido cuyos nodos son esquemas (de acción) y cuyos enlaces representan la relación "derivada de". Los esquemas primitivos (vacíos) incorporados que controlan directamente los efectores, son análogos a los conjuntos de células motoras que controlan los grupos musculares en humanos. Un esquema consiste en una acción, junto con su contexto y su resultado. En la periferia del esquema, se encuentran los esquemas vacíos (esquemas con una acción simple, pero sin contexto o resultados), mientras que los esquemas más complejos que consisten en acciones y secuencias de acciones se descubren a medida que uno se mueve hacia adentro. Para que un esquema actúe, primero debe crearse una instancia y luego seleccionarse para su ejecución de acuerdo con el mecanismo de selección de acción.

5. Conciencia funcional

El módulo de "conciencia" de LIDA implementa los procesos de la teoría de Global Workspace (Baars 1988) mediante *codelets*. Estos están especializados para alguna tarea simple y, a menudo, desempeñan el papel de observador de un *demonio (daemon)* para una condición adecuada bajo la cual actuar. El aparato para la "conciencia" funcional consiste en un gestor de coalición, un controlador de centro de atención, un administrador de transmisión y códigos de atención que reconocen situaciones novedosas o problemáticas.

6. Selección de acción

La arquitectura LIDA emplea una mejora de la red de comportamiento de Maes (1989) para la selección de acciones de alto nivel al servicio de los sentimientos y las emociones. Varios sentimientos y emociones distintos operan en paralelo, tal vez variando en urgencia a medida que pasa el tiempo y el entorno cambia. La red de comportamiento es un dígrafo (gráfico dirigido) compuesto por comportamientos (esquemas de acción instanciados) y sus diversos enlaces. Como en los modelos conexionistas (redes neuronales), este dígrafo difunde la activación. La activación proviene de cuatro fuentes: de la activación preexistente almacenada en los comportamientos, del entorno, de los sentimientos y emociones y de los estados internos. Para que se active, un comportamiento debe ser ejecutable, debe tener una activación por encima del umbral definido y debe tener la mayor activación.

Antes de describir el ciclo cognitivo de LIDA se muestra a continuación un diagrama del modelo de LIDA para su mayor comprensión.

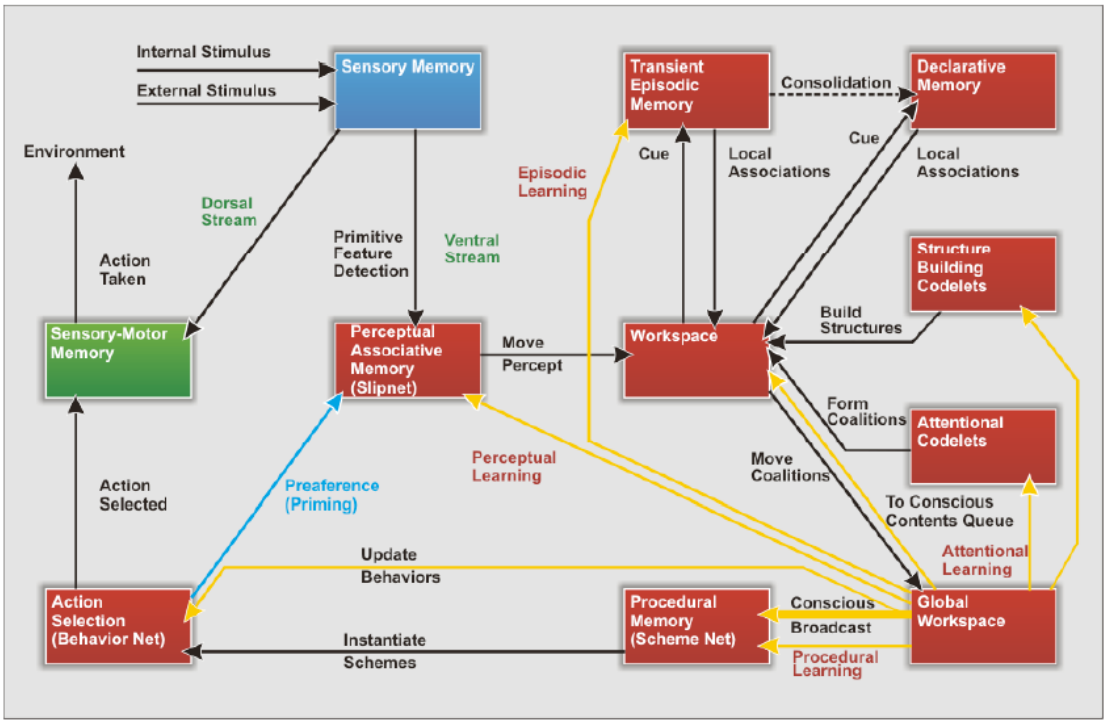

Figura 1 – Diagrama del modelo de LIDA – Fuente: [33]

4.2.1.3 Ciclo cognitivo

Todo agente autónomo dentro de un ambiente complejo y dinámico debe frecuentemente sensar y actuar dentro de él iterativamente, en lo que se conoce como ciclo cognitivo. Ciclos de actividad en serie, pero superpuestos, que generalmente comienzan en la percepción y terminan en una acción. Se piensa que ocurren entre 5 y 10 ciclos cognitivos por segundo en los humanos, en cascada, para que algunos de los pasos en ciclos adyacentes se produzcan en paralelo. La serialización se conserva en las emisiones conscientes. A continuación, se describe el ciclo cognitivo dividiéndolo en nueve pasos.

1. Percepción

   Los estímulos sensoriales, externos o internos, son recibidos e interpretados por la percepción que produce los comienzos del significado.

2. Percepción del búfer preconsciente

   La percepción, incluidos algunos de los datos más el significado, así como las posibles estructuras relacionales, se almacenan en los búferes preconscientes de la memoria de trabajo de LIDA (espacio de trabajo). Aquí se construyen estructuras temporales.

3. Asociaciones locales

   Usando la percepción entrante y los contenidos residuales de la memoria de trabajo, incluido el contenido emocional, como claves, las asociaciones locales se recuperan automáticamente de la memoria episódica transitoria y de la memoria declarativa, y se almacenan en la memoria de trabajo a largo plazo.

4. Competición por consciencia

   Los *codelets* de atención miran la memoria de trabajo a largo plazo y traen a la conciencia eventos nuevos, relevantes, urgentes o insistentes.

5. Transmisión consciente

   Una coalición de *codelets*, típicamente un *codelet* de atención y su conjunto de *codelets* de información relacionada que llevan contenido, obtiene acceso al espacio de trabajo global y sus contenidos se transmiten. En los seres humanos, se presume que esta transmisión corresponde a la conciencia fenoménica.

6. Reclutamiento de recursos

   Los esquemas relevantes responden a la emisión consciente. Estos son típicamente esquemas cuyo contexto es relevante para la información en la transmisión consciente. Así, la conciencia resuelve el problema de relevancia en el reclutamiento de recursos.

7. Establecimiento de jerarquía de contextos de objetivos

   Los esquemas reclutados utilizan los contenidos de la conciencia, incluidos los sentimientos / emociones, para instanciar las nuevas jerarquías de contexto de objetivos (copias de sí mismos) en la red de comportamiento, vincular sus variables y aumentar su activación. Otras condiciones ambientales determinan cuál de los contextos de objetivos anteriores también recibe vinculación variable y/o activación adicional.

8. Acción elegida

La red de comportamiento elige un solo comportamiento (esquema, contexto de objetivo), de un flujo de comportamiento recientemente instanciado o previamente activo. Cada selección de un comportamiento incluye la generación de un *codelet* de expectativa.

9. Acción tomada

La ejecución de un comportamiento (contexto de objetivos) da como resultado que los *codelets* de comportamiento realicen sus tareas especializadas, tengan consecuencias externas o internas, o ambas cosas. LIDA está tomando una acción. Los *codelets* que actúan también incluyen al menos un *codelet* de expectativa cuya tarea es monitorear la acción, dando a conocer cualquier falla en los resultados esperados.

4.2.1.4 Procesos multi-cíclicos

Los procesos cognitivos de orden superior, como el razonamiento, la resolución de problemas, la imaginación, etc., en LIDA ocurren a lo largo de múltiples ciclos cognitivos. A continuación, se describe la deliberación, la acción voluntaria, la resolución no rutinaria de problemas y la automatización como algunos de los procesos multi-cíclicos que provee LIDA. Los mecanismos que realizan estos procesos se implementan normalmente como flujos de comportamiento en la memoria de procedimientos.

1. Deliberación

Cuando los humanos necesitan resolver un problema, usualmente crean en la mente diferentes estrategias o posibles soluciones. De esta manera, se simulan los efectos de ejecutar cada estrategia o soluciones de prueba sin realmente hacerla. Esto es similar a simular una realidad virtual interna. Eventualmente, se decide por sobre una estrategia o solución de prueba para resolver el problema utilizándola. Este proceso se llama deliberación. Durante el proceso de deliberación, muchas ideas potencialmente conflictivas compiten para ser seleccionadas como una estrategia o solución del problema. Una de ellas es elegida voluntariamente. Deliberación en LIDA es implementada utilizando información consciente para crear escenarios y evaluar sus utilidades.

2. Acción voluntaria

Acciones voluntarias involucran una deliberación consciente en la decisión de tomar una acción. LIDA suministra un mecanismo subyacente que implementa la teoría ideomotora basada en la voluntad.

Los participantes en este proceso de toma de decisiones incluyen proponer y objetar *codelets de atención* y un *codelet cronometrador*. Una tarea puede proponer un *codelet de atención*, de esta manera se propone una determinada acción en función de su patrón particular de preferencias. El *codelet* de atención que propone aporta información sobre sí mismo y la acción propuesta a la "conciencia", de modo que si no hay ningún otro objeto *codelet* de atención objetante (al hacerse "consciente" con un mensaje de objetar), y si no hay otro *codelet* de atención que propone, hace una propuesta diferente. Dentro de un lapso de tiempo dado, el *codelet* del cronometrador decidirá sobre la acción propuesta. Si se hace una objeción o una nueva propuesta de manera oportuna, el *codelet* del cronometrador detiene el tiempo o restablece el tiempo para la nueva propuesta.

Los participantes en el proceso de toma de decisiones son: *codelet* de atención que pueden realizar dos acciones: proponer u objetar, y *codelet* cronometrador que controla el tiempo de la toma de decisión.

3. Resolución no rutinaria de problemas

   Con la ayuda de su mecanismo de conciencia, LIDA tiene la capacidad de lidiar con casos nuevos de situaciones rutinarias. Sin embargo, para manejar eficientemente situaciones novedosas, problemáticas inesperadas, el modelo necesita alguna forma de resolución de problemas no rutinaria. En general, la resolución de problemas no rutinarios se refiere a la capacidad de idear soluciones para situaciones problemáticas novedosas. Este tipo de solución generalmente se denomina *malla*, donde los humanos utilizan fragmentos de conocimientos previos para obtener soluciones a problemas. La resolución de problemas no rutinarios es bastante similar a la planificación en la IA clásica. Sin embargo, si bien la planificación en la IA clásica supone que todos los operadores individuales están continuamente disponibles para su consideración, no se hace tal suposición debido a su inverosimilitud cognitiva. En cambio, el enfoque se basa en la conciencia para reclutar piezas de conocimiento inconscientes que son potencialmente relevantes para la solución. La resolución de problemas no rutinarios en la arquitectura LIDA se ve mejor como un flujo de comportamiento único que opera en múltiples ciclos, con la configuración de planes de acción parciales en cada ciclo.

4. Automatización

   La automatización se refiere a la capacidad humana (y animal) de aprender una tarea de procedimiento en la medida en que la tarea se puede realizar sin una intervención consciente. Dado que la conciencia es un recurso limitado, las tareas automatizadas liberan este recurso para actividades cognitivas más apremiantes, como la deliberación, la resolución de problemas, el razonamiento, etc. En la arquitectura LIDA, los planes parciales de acciones se representan mediante flujos de comportamiento (jerarquías de contexto de objetivos que consisten en comportamientos que operan aproximadamente en una secuencia). Para una tarea no automatizada, se requiere conciencia para reclutar para la ejecución del siguiente comportamiento en una secuencia instanciada. La automatización se implementa en LIDA por medio de comportamientos en una corriente que crea automáticamente asociaciones entre sí, eliminando así la necesidad de una intervención consciente. Una vez que una tarea se automatiza, la ejecución de los comportamientos individuales se supervisa mediante *codelets* de expectativa. Cuando se observa una ejecución fallida y se toma conciencia de esta información, se contrata el proceso de des-automatización para suspender temporalmente la automatización, restaurando así la intervención consciente.

4.2.1.5 Desarrollo de aprendizaje en LIDA

LIDA posee tres tipos de mecanismos de aprendizaje fundamentales que subyacen del aprendizaje humano. Aprendizaje perceptual, el aprendizaje de nuevos objetos, categorías, relaciones, etc. Aprendizaje episódico de eventos, de qué, dónde y cuándo. Y por último aprendizaje procedural de aprendizaje de nuevas acciones. Aunque el tipo de conocimiento retenido debido a estos tres mecanismos de aprendizaje difiere, los mecanismos están fundados en dos premisas básicas. La primera premisa denota que conciencia consciente es

suficiente para el aprendizaje. La segunda premisa, que es compartida entre varios mecanismos de aprendizaje, es que el aprendizaje es modulado por sentimientos y emociones.

El desarrollo del aprendizaje en LIDA ocurre durante la transmisión consciente. [32]

4.2.1.6 LIDA Framework

LIDA posee un framework desarrollado en Java el cual implementa los componentes principales de su modelo, *LIDA Model*, y permite poder añadir o acoplar los módulos que sean necesarios a éste para adaptarlos a las necesidades del proyecto que se quiera desarrollar [16]. Es de software libre siempre y cuando no se lo use con fines comerciales. Su investigación y soporte está a cargo del *Cognitive Computing Research Group de la universidad de Memphis* [17].

4.2.2 - COG PRIME

4.2.2.1 CogPrime and OpenCog

CogPrime está estrechamente relacionado con el framework de IA de código abierto *OpenCog*. Pero los dos no son sinónimos. OpenCog es un framework más general, adecuado para la implementación de una variedad de aplicaciones de IA especializadas, así como, potencialmente, diseños alternativos de AGI. CogPrime podría implementarse potencialmente fuera del framework de OpenCog. La implementación particular de CogPrime en OpenCog se llama OpenCog Prime. OpenCog fue diseñado con el propósito, junto con otros, de permitir una implementación eficiente y escalable del diseño completo de CogPrime.

4.2.2.2 Contexto filosófico

Tener una filosofía de la mente apropiada, ciertamente no es garantía de crear un sistema de AGI avanzado; La filosofía sola, va mucho más allá de la implementación. Sin embargo, tener una filosofía de la mente inapropiada puede ser una gran barrera en la creación de sistemas de AGI.

El desarrollo del diseño CogPrime se ha guiado sustancialmente por una filosofía de la mente llamada *patternism* (basada en patrones [5]). Debido al papel central que ha desempeñado el *patternism* en el desarrollo de CogPrime, comprender algunas cosas sobre la filosofía generalista es útil para comprender CogPrime, incluso para aquellos que no tienen una inclinación filosófica.

La filosofía de la mente "*patternism*" es un enfoque general para pensar en sistemas inteligentes. Se basa en la premisa muy simple de que la mente está hecha de patrones, y que una mente es un sistema para reconocer patrones, tanto en sí misma como en el mundo, que incluye críticamente patrones con respecto a qué procedimientos es probable que conduzcan al logro de qué objetivos en qué contextos.

En esta perspectiva, el "patrón" se define generalmente como "representación como algo más simple".

Básicamente, la mente de un sistema es el conjunto difuso de diferentes representaciones simplificadoras de ese sistema que pueden adoptarse.

La inteligencia se concibe como la capacidad de lograr objetivos complejos en entornos complejos; donde la complejidad en sí misma puede definirse como la posesión de una rica variedad de patrones. Una mente es, por lo tanto, una colección de patrones que se asocia con un proceso dinámico persistente que logra objetivos con gran cantidad de patrones en entornos con gran cantidad de patrones.

Una hipótesis adicional hecha dentro de esta filosofía es que la reflexión es crítica para la inteligencia. Esto nos permite concebir un sistema inteligente como un sistema dinámico que reconoce patrones en su entorno y en sí mismo, como parte de su búsqueda para lograr objetivos complejos.

Entre los muchos tipos de patrones en los sistemas inteligentes, los patrones semióticos son particularmente interesantes. Se descomponen en tres categorías:

Patrones icónicos, que son patrones de similitud interna contextualmente importante entre dos entidades

Patrones indexados, que son patrones de co-ocurrencia espacio-temporal

Patrones simbólicos, que son patrones que indican que dos entidades a menudo están involucradas en las mismas relaciones.

La búsqueda de esta filosofía, en detalles, conduce a una variedad de hipótesis y conclusiones particulares sobre la naturaleza de la mente. A partir de la visión de la inteligencia en términos de alcanzar objetivos complejos en entornos complejos, llega una visión en la que se entiende que la dinámica de un sistema cognitivo se rige por dos fuerzas principales:

Auto-organización, a través de la cual la dinámica del sistema hace que los patrones existentes del sistema den lugar a otros nuevos.

Comportamiento orientado a objetivos, que básicamente equivale a un sistema que interactúa con su entorno de una manera que parece un intento de maximizar alguna función simple razonable.

A estas fuerzas principales se las debe entenderse como aspectos cooperativos.

4.2.2.2.1 Un principio de correspondencia mente-mundo

Un principio filosófico adicional guío el diseño de CogPrime, este es el principio de correspondencia mente-mundo, el cual ensancha la noción de inteligencia como adaptación a ambientes.

Mentes del mundo real están siempre adaptadas a ciertas clases de ambientes y objetivos, incluso sistemas de una gran cantidad de inteligencia general, sujetas a las restricciones de tiempo y espacio del mundo real, serán necesariamente más eficientes a ciertos tipos de aprendizajes que otros.

Para que la inteligencia ocurra, tiene que haber una correspondencia natural entre la transición de secuencias de los estados del ambiente y las correspondientes transiciones de secuencias de los estados de la mente, al menos en los casos de las transiciones de secuencias que conducen a objetivos relevantes.

4.2.2.3 Arquitectura de alto nivel de CogPrime

CogPrime no se ha derivado directamente de esos principios filosóficos; más bien, se ha creado comenzando con una combinación de algoritmos y estructuras de psicología cognitiva humana y ciencias de la computación, y luego formando esta combinación para producir un sistema que parece ser compatible con estos principios filosóficos, además de ser computacionalmente factible en hardware actual y que contiene estructuras cognitivas y dinámicas aproximadamente homólogas a las principales estructuras humanas.

Un principio subyacente clave es: el uso de múltiples procesos cognitivos asociados con múltiples tipos de memoria para permitir que un agente inteligente ejecute los procedimientos que cree que tienen la mejor probabilidad de funcionar hacia sus objetivos en su contexto actual.

4.2.2.4 Representación local y global del conocimiento

Una de las decisiones más importantes que se debe tomar al diseñar un sistema AGI es cómo el sistema debe representar el conocimiento. Naturalmente, cualquier sistema AGI avanzado sintetizará muchas de sus propias representaciones de conocimiento para manejar tipos particulares de conocimiento, pero, aun así, un diseño AGI generalmente hace al menos algún tipo de compromiso sobre la categoría de mecanismos de representación de conocimiento hacia los cuales el sistema AGI será sesgado. Los mecanismos de representación del conocimiento de OpenCog se basan fundamentalmente en redes. Esto se debe a que, a nivel filosófico, una de las metáforas más poderosas conocidas para comprender las mentes es verlas como redes de elementos interrelacionados e interconectados.

Las dos mayores súper categorías de representación del conocimiento son la "local" (también llamada explícita) y "global" (también llamada implícita), con una categoría híbrida que refiera a "glocal", la cual combina ambas. Los tres tipos de representaciones del conocimiento pueden ser realizados por redes. En CogPrime, los tres son realizados con la misma red (Atomspace).

4.2.2.4.1 Hipergrafos etiquetados y con pesos

Hay muchos mecanismos para la representación del conocimiento en sistema de AI en una forma explícita y localizada, en su mayoría descienden de varias variantes de lógica formal.

En la superficie, el esquema de representación de CogPrime no es tan diferente de muchos otros enfoques. Sin embargo, las particularidades del conocimiento explícito de CogPrime, es que están configurados cuidadosamente para corresponderse con los procesos cognitivos de CogPrime, los que son los más distintivos en naturaleza de otros mecanismos de representación.

Un hipergrafo es una estructura matemática abstracta, que consiste en objetos llamados nodos y objetos llamados links que conectan estos nodos. Un grafo tradicionalmente significa un conjunto de puntos conectados con líneas. Un hipergrafo, por otro lado, puede tener múltiples links que conectan más de dos nodos.

En CogPrime es más útil considerar hipergrafos generalizados que extienden los hipergrafos ordinarios agregando dos características adicionales:

1. Links que apuntan a links en vez de a nodos
2. Nodos que, cuando se mira más en detalle, contienen hipergrafos embebidos.

Un término genérico que engloba a Links y Nodos es Atom.

Un hipergrafo ponderado y etiquetado es un hipergrafo cuyos links y nodos contienen etiquetas y uno o más números que son generalmente llamados "pesos". Una etiqueta asociada a un link o un nodo puede algunas veces ser interpretado como que nos dice qué tipo de entidades, o alternativamente que nos dice qué tipo de dato tiene asociado el Nodo. Por otro lado, un ejemplo de un peso que puede estar adjunto a un link o a un nodo es un número que representa una probabilidad, o un número que representa cuán importante es un nodo o un link.

Obviamente, hipergrafos vienen de varios tipos de dinámicas. Mínimamente, uno puede pensar en:

1. Dinámicas que modifican las propiedades de los nodos o links en el hipergrafo
2. Dinámicas que agregan nuevos nodos o links a un hipergrafo, o remueven existentes.

Ambos tipos de dinámicas son muy importantes en CogPrime.

4.2.2.4.2 Memoria "Glocal"

La coordinación "*Glocal*" de memoria local y memoria global juega un rol importante en la arquitectura de OpenCog. Una memoria "*glocal*" es una que trasciende la dicotomía global/local e incorpora ambos aspectos. Se puede considerar como una memoria distribuida. En un sistema de memoria *glocal*, la mayoría de los elementos de la memoria son almacenados en ambas maneras, local y globalmente, con la propiedad de que obtener cualquiera de los dos registros de un elemento también tiende a generar el otro.

La memoria *glocal* aplica a múltiples formas de memoria, sin embargo, se enfoca largamente en memoria perceptual y declarativa.

La idea central de una memoria *glocal* es que, elementos pueden ser almacenados en memoria en una forma de estructuras emparejadas llamadas pares (clave, mapa). La clave es la versión localizada del elemento, y guarda algún aspecto significante de los elementos en una forma simple y limpia. El mapa es una versión dispersa y distribuida del elemento, que representa el elemento como una combinación de fragmentos de otros elementos. El mapa incluye la clave como un subconjunto; la activación de la clave generalmente causa la activación del mapa; y cambios en elemento de memoria generalmente involucrarán cambios complejos coordinados en ambos, la clave y el mapa.

La memoria es un área donde la arquitectura del cerebro animal difiere radicalmente de una arquitectura de Von Neumann, arquitectura subyacente en casi todas las computadoras contemporáneas. Las computadoras de Von Neumann separan la memoria del procesamiento, mientras que en el cerebro humano no existe tal distinción. La memoria humana está generalmente construida en forma de recuerdos, lo que le da a la memoria humana una gran capacidad de "llenar los agujeros" de la experiencia recordada y el conocimiento; y también causa problemas con los recuerdos inexactos en muchos contextos. Se cree que un aspecto constructivo de la memoria es largamente asociado con esta característica global/local.

4.2.2.5 Tipos de memorias y procesos cognitivos asociados

Los diagramas de arquitecturas están generalmente bien, pero, en última instancia sus dinámicas hacen que una arquitectura tome vida. La inteligencia es todo relacionado con el aprendizaje, que está por definición atada al cambio, en cuanto a la respuesta dinámica del ambiente y la auto-organización de las dinámicas internas.

CogPrime confía en múltiples tipos de memorias, y fundada en la premisa que el camino correcto en construir una AGI pragmática y similar a la humana, es manejar diferentes tipos de memorias de formas diferentes en términos de estructuras y dinámicas.

4.2.2.6 Dinámicas orientadas a objetivos

La dinámica básica de la dinámica orientada a objetivos del sistema CogPrime, dentro de la cual varios tipos de memorias son utilizadas, es dirigida por implicaciones conocidas como "semánticas cognitivas" que tienen la forma de:

Contexto ^ *Procedimiento* -> *Goal* <*p*>

En forma resumida se puede expresar como C ^ P -> G. Semi-formalmente, esta implicación puede ser interpretada como: "Si el contexto "C" parece mantenerse actualmente, entonces si se utiliza el procedimiento P se puede esperar alcanzar el objetivo G con una cierta certeza representada por el valor de verdad del objeto "P".

El esquema cognitivo de CogPrime es significativamente similar a reglas de producción en arquitecturas clásicas como SOAR y ACT-R, sin embargo, hay también diferencias significativas que son importantes para las funcionalidades de CogPrime. A diferencia de los sistemas de reglas de producción clásicos, incertidumbre es clave para la representación de conocimiento de CogPrime, y cada esquema cognitivo es nombrado con una incertidumbre de valor de verdad, la cual es crítica para su utilización en el proceso cognitivo. Además, esquemas cognitivos pueden estar incompletos, faltando uno o dos términos, que pueden ser completados por varios procesos cognitivos.

Finalmente, la diferencia más grande entre los esquemas cognitivos de CogPrime y las reglas de producción u otras construcciones similares, es que en CogPrime este nivel de representación del conocimiento no es la única importante. CLARION, es un ejemplo de arquitectura cognitiva que usa reglas de producción para representación de conocimiento explícito y luego usa un almacenamiento de conocimiento simbólico totalmente separado para el conocimiento implícito. En CogPrime, ambos conocimientos, explícito e implícito son almacenados en el mismo grafo de nodos y links.

4.2.2.6.1 Procesos de análisis y síntesis

El esquema cognitivo *Contexto* ^ *Procedimiento* -> *Goal* <*p*> lleva a la conceptualización de la acción interna de un sistema inteligente involucrando dos categorías importantes de aprendizaje.

Análisis: Estimando la probabilidad de "p" de una relación postulada C ^ P -> G.

Síntesis: Rellenando una o dos variables del esquema cognitivo, dadas suposiciones relacionadas a las variables restantes, y dirigidas para el objetivo de maximizar la probabilidad del esquema cognitivo.

4.2.2.7 Afirmaciones claves

4.2.2.7.1 Sistemas multi-memoria

La primera afirmación es que, para alcanzar inteligencia general en el contexto de inteligencia humana, ambientes amigables y objetivos, usando recursos computacionales factibles, es importante que el sistema de AGI pueda manejar diferentes tipos de memoria. (declarativa, procedural, episódica, sensorial, intencional, atencional) de formas personalizables pero interoperables. La idea básica es que estos diferentes tipos de conocimiento tienen características muy diferentes, por lo que tratar de manejarlas a todas en un solo enfoque, aunque posible, es probablemente inaceptablemente ineficiente.

En CogPrime se utiliza una combinación compleja de representaciones, incluido el Atomspace, para conocimiento declarativo, conocimiento atencional e intencional y algunos conocimientos episódicos y sensorio-motores, programas combinados para conocimiento procesal, simulaciones para conocimiento episódico y redes neuronales jerárquicas para algunos sensores conocimiento motor.

En los casos en que se utiliza el mismo mecanismo de representación para diferentes tipos de conocimiento, se utilizan diferentes procesos cognitivos y, a menudo, diferentes aspectos de representación.

Pragmáticamente, también está bastante claro que el cerebro humano adopta un enfoque de memoria múltiple, por ejemplo, con el cerebelo y las regiones corticales vinculadas estrechamente que contienen estructuras especiales para manejar el conocimiento del procedimiento, con estructuras especiales para manejar factores motivacionales (intencionales), etc. Y, décadas de ciencia de la computación y práctica de inteligencia artificial estrecha sugieren fuertemente que el enfoque de "una estructura de memoria para todos" no es capaz de conducir a enfoques efectivos en el mundo real.

4.2.2.7.2 Percepción acción y ambiente

Cuanto más entendemos la inteligencia humana, más claro se vuelve qué tan cerca ha evolucionado para coincidir con los objetivos y entornos particulares para los que evolucionó el organismo humano. Esto es cierto en un sentido amplio, como lo ilustran los problemas anteriores con respecto a los sistemas de memoria múltiple, y también es cierto en muchos detalles, como se ilustra, por el análisis evolutivo de Changizi [Cha09] del sistema visual humano.

4.2.2.7.3 Representación del conocimiento

Dadas las fortalezas y debilidades de las computadoras digitales actuales y futuras, una red (netamente simbólica) es una buena representación para almacenar directamente muchos tipos de memoria e interactuar entre aquellos que no almacena directamente. AtomSpace de CogPrime es una red neural-simbólica diseñada para funcionar bien con PLN, MOSES, ECAN y otros procesos cognitivos clave de CogPrime; les proporciona lo que necesitan sin causarles complejidades indebidas. Proporciona una plataforma que estos procesos cognitivos pueden utilizar para construir de manera adaptativa y automática

representaciones de conocimiento especializado para tipos particulares de conocimiento que encuentran.

4.2.2.7.4 Procesos cognitivos

El quid de la inteligencia es la dinámica, el aprendizaje, la adaptación; y entonces el quid de un diseño AGI es el conjunto de procesos cognitivos que proporciona el diseño. Estos procesos deben permitir colectivamente que el sistema AGI logre sus objetivos en sus entornos utilizando los recursos disponibles. Dado el diseño de memoria múltiple de CogPrime, es natural considerar los procesos cognitivos de CogPrime en términos de en qué subsistemas de memoria se enfocan.

4.2.2.7.5 Completando la ecuación cognitiva

Una afirmación clave basada en la noción de "Ecuación cognitiva es que es importante que un sistema inteligente tenga alguna forma de reconocer patrones de gran escala en sí mismo, y luego encarnar estos patrones como nuevos, elementos de conocimiento localizados en su memoria. Esta dinámica introduce una dinámica de retroalimentación entre el patrón emergente y el sustrato, que se presume que es crítica para la inteligencia general bajo recursos computacionales factibles. También se relaciona muy bien con la noción de memoria "glocal" que esencialmente plantea una localización de algunos recuerdos globales, lo que naturalmente dará como resultado la formación de algunos recuerdos glocales. Una de las ideas clave que subyace en el diseño de CogPrime es que, dado el uso de una red neural-simbólica para la representación del conocimiento, una "heurística" de formación de mapas basada en minería de grafos es una buena manera de hacerlo.

4.2.2.7.6 Sinergia cognitiva

Las sinergias son absolutamente críticas para la funcionalidad propuesta del sistema CogPrime. Sin ellos, los mecanismos cognitivos no funcionarán adecuadamente, sino que sucumbirán a las explosiones combinatorias. Los otros aspectos de CogPrime (la arquitectura cognitiva, la representación del conocimiento, el marco de realización y la metodología de enseñanza del desarrollo asociada) también son críticos, pero ninguno de ellos generará el surgimiento crítico de la inteligencia sin mecanismos cognitivos que escalen efectivamente. Y, en ausencia de mecanismos cognitivos que escalen efectivamente por sí mismos, debemos confiar en los mecanismos cognitivos que efectivamente se ayudan mutuamente a escalar. [18]

4.2.3 - CLARION

CLARION es un proyecto que investiga las estructuras fundamentales de la mente humana. En particular, intenta explorar la interacción de la cognición implícita y explícita, enfatizando el "bottom-up learning" (aprendizaje de abajo hacia arriba). El proyecto está dirigido a la síntesis de muchas ideas intelectuales en un modelo coherente de cognición. El objetivo es crear una arquitectura genérica que contenga una variedad de procesos cognitivos de una manera unificada y así proporcionar explicaciones unificadas de una amplia gama de datos [19].

En general CLARION es un modelo integrador que consiste en varios subsistemas funcionales y además en una estructura de representación dual entre representaciones implícitas y explícitas siendo dos componentes separados en cada uno de los subsistemas. [20]

Los subsistemas que componen la arquitectura son Action-centered subsystem (ACS, Subsistema centrado en acciones), Non-action-centered subsystem (NACS, Subsistema no centrado en acciones), motivational subsystem (MS, subsistema motivacional) y metacognitive subsystem (MCS, subsistema metacognitivo).

El rol del ACS es de controlar las acciones, independientemente de si las acciones son movimientos externos u operaciones mentales internas. El rol del NACS es de dar mantenimiento al conocimiento general tanto implícito como explícito. El rol del MS es de proveer las motivaciones subyacentes para la percepción, acción y cognición en términos de proveer ímpetu y retroalimentación. El rol del MCS es de monitorear, dirigir y modificar las operaciones del ACS dinámicamente como así también las operaciones de los demás subsistemas.

Cada uno de estos subsistemas contiene dos niveles de representación. Generalmente un nivel superior que codifica un conocimiento explícito y un nivel inferior que codifica un conocimiento implícito. [21]

En la figura 2 se muestra una imagen de la arquitectura CLARION y las interacciones de sus subsistemas:

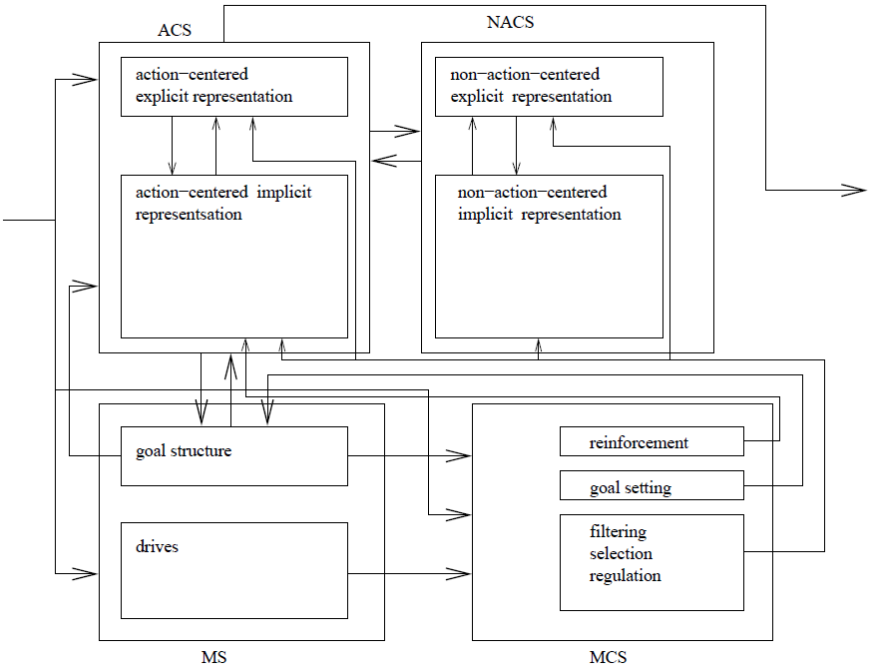

Figura 2 - Arquitectura CLARION – Fuente [19]

A continuación, se describen brevemente cada uno de los subsistemas:

4.2.3.1 - Subsistema centrado en acciones

El subsistema centrado en acciones (ACS) está destinado a la toma de decisiones de un agente cognitivo individual en su interacción con el ambiente. Este subsistema es el más importante de CLARION. Esencialmente el proceso de toma de decisiones es el siguiente: Observando el estado actual del ambiente, los dos niveles de procesos dentro del ACS toman sus propias decisiones de forma separada de acuerdo con su propio conocimiento y luego el resultado es de alguna manera una combinación de ambos. Luego, se toma la selección final de la acción y esta es ejecutada. De alguna manera, el ambiente es modificado por esta acción. Comparando el nuevo estado del ambiente con el estado previo, el agente aprende. Este es un ciclo que luego se repite.

Dentro del subsistema, el nivel inferior es denominado IDNs (Implicit decisión networks o redes de decisión implícita), que es implementado con redes neuronales involucrando representaciones distribuidas, y por otro lado el nivel superior es denominado ARS (Action Rule Store o almacenamiento de reglas de acciones), implementado usando representaciones simbólicas/localistas.

En el nivel inferior, la entrada *x* (estado actual del ambiente) consiste en tres conjuntos de información: (1) entradas sensoriales, (2) elementos de memoria de trabajo, (3) elemento seleccionado de la estructura de objetivos. Estos tres son divididos en un conjunto de dimensiones, y así el estado *x* es representado como un conjunto de pares dimensión-valor: *(d1, v1) (d2, v2)... (dn,vn).*

La salida de este nivel inferior es una elección de acción, la cual consiste en tres grupos de acciones: acciones de memoria de trabajo, acciones de objetivos y acciones

externas. En cada una de las redes, las acciones son seleccionadas basadas en sus valores. Estos valores son una evaluación de cuán deseable es una acción *a* dado un estado *x* del ambiente, *Q(x,a).* Luego, esta evaluación de calidad es usada para decidir probabilísticamente sobre una acción para ser realizada, a través de una distribución de Boltzmann.

Luego, el conocimiento explícito en el nivel superior (ARS) es capturado por reglas y trozos (*rules and chunks*). Las condiciones de una regla, similarmente a las entradas del nivel inferior, consisten en tres grupos de información: información sensorial, elementos de memoria de trabajo, objetivo actual. La salida de una regla, es una selección de acción. La condición de una regla constituye una entidad distinta conocida como *chunk* (trozo); y esa es la conclusión de una regla.

Específicamente, las reglas tienen esta forma: *especificación-de-estado →acción*. La parte izquierda (la condición) es una conjunción de elementos individuales. Cada uno de los elementos refiere a una dimensión del estado x del ambiente. La parte derecha (la conclusión) es una recomendación de acción.

Cada valor de cada dimensión de estado es representado por un nodo individual en el nivel inferior. Estos nodos en el nivel inferior están conectados con un único nodo en el nivel superior representando esa condición, conocido como "*chunk node*" (una representación localista). Para capturar este proceso de aprendizaje desde abajo hacia arriba, el algoritmo "Rule-Extraction-Refinement" aprende reglas en el nivel superior utilizando información del nivel inferior. La idea básica de este aprendizaje desde abajo hacia arriba es la siguiente: Si una acción seleccionada (por el nivel inferior) es exitosa (satisface algún criterio), entonces una regla explícita es extraída en el nivel superior. Luego, en las posteriores interacciones con el ambiente, la regla es refinada considerando el resultado obtenido de aplicar la regla. Si esta es exitosa la regla es generalizada para hacerla más universal. Si no es exitosa, la regla se hace más específica y exclusiva al caso particular.

Un agente necesita una base racional para tomar las decisiones mencionadas anteriormente. Para ello un criterio numérico es implementado, y así medir un resultado exitoso o no. Esencialmente, durante la selección de reglas en cada paso, conteos de correspondencias positivas y negativas son actualizadas mediante la medición de sí una regla potencialmente llevará a un resultado positivo o negativo. Luego basado en ello, las mediciones se comparan entre las diferentes reglas y se eligen las que mejores resultados obtuvieron.

Por otra parte, esta representación dual (implícita y explícita) también permite un aprendizaje de arriba hacia abajo. Con conocimiento explícito (en forma de reglas) en el nivel superior, el nivel inferior aprende bajo la guía de las reglas. Esto significa que inicialmente el agente confía mayormente en las reglas del nivel superior para la toma de decisiones, pero luego, gradualmente a medida que más y más conocimiento es adquirido por el nivel inferior comienza a confiar más en este último. [22]

4.2.3.2 - Subsistema no centrado en acciones

El subsistema no centrado en acciones (NACS) es usado para la representación general del conocimiento acerca del ambiente que no está focalizado en acciones, con el propósito de realizar inferencias acerca del mismo. Este conocimiento es almacenado también como una representación dual, como lo era en el ACS: en la forma explícita reglas asociativas (en el nivel superior) y en la forma implícita memoria asociativa (en el nivel inferior).

En el nivel inferior, redes de memoria asociativa (AMNs, associative memory networks) codifican el conocimiento implícito. Asociaciones son creadas estableciendo relaciones entre una entrada y una salida. El algoritmo de aprendizaje por retro propagación puede ser utilizado para establecer este tipo de asociaciones.

Por otro lado, en el nivel superior, almacenamiento de conocimiento general (GKS, general knowledge store) codifica el conocimiento explícito.

La forma básica de un *chunk* consiste en un *chunk* id y un conjunto de pares dimensión-valor. Un nodo es configurado en el GKS para representar un *chunk*. Un nodo *chunk* conecta con sus características constituyentes (por ejemplo, pares dimensión-valor) representadas como nodos individuales en el nivel inferior. Adicionalmente, conexiones entre *chunks* codifican asociaciones explícitas entre pares de nodos *chunk*, que son conocidas como reglas asociativas. Estas podrán ser aprendidas y creadas de múltiples maneras.

Por encima de ello, razonamiento basado en similitudes puede ser empleado en NACS. Un *chunk* puede ser comparado con otro y si la similitud es suficientemente alta el segundo es inferido.

Como en ACS, el aprendizaje de desde abajo hacia arriba o desde arriba hacia abajo pueden ser utilizados, tanto para extraer conocimiento explícito en el nivel superior desde el conocimiento implícito del nivel inferior o para asimilar el conocimiento explícito del nivel superior en conocimiento implícito del nivel inferior. [23]

4.2.3.3 - Subsistema motivacional

Los procesos de supervisión del ACS y NACS se componen de dos subsistemas. El subsistema motivacional y el subsistema meta-cognitivo. El subsistema motivacional se ocupa de las unidades y sus interacciones. Eso significa que se preocupa de porqué un agente hace lo que hace, porqué un agente toma la acción que toma. Simplemente diciendo que un agente decide que acción tomar para maximizar ganancias, recompensas o pagos deja abierta la puerta a la pregunta de qué determina ganancias, recompensas o pagos. La relevancia del subsistema motivacional a la principal parte de la arquitectura, el ACS, cae principalmente en el hecho de que provee el contexto en el que el objetivo y la retroalimentación del ACS son determinados. De este modo influye en el funcionamiento del ACS, y por extensión, el funcionamiento del NACS.

Por un lado, desde hace ya varias décadas, las críticas a los modelos comúnmente aceptados de motivaciones humanas, por ejemplo, en economía, se han centrado en sus puntos de vista demasiado estrechos con respecto a las motivaciones, por ejemplo, únicamente en términos de recompensas económicas simples y castigos.

Muchos críticos se opusieron a la aplicación de este enfoque demasiado estrecho en las ciencias sociales, del comportamiento, cognitivas y políticas. Las motivaciones sociales complejas, como el deseo de reciprocidad, la búsqueda de aprobación social y el interés por la exploración, también tienen el comportamiento humano. Al descuidar estas motivaciones, la comprensión de algunos problemas sociales y de comportamiento clave (como el efecto de los incentivos económicos en el comportamiento individual) puede verse obstaculizada. Se pueden aplicar críticas similares al trabajo sobre el aprendizaje por refuerzo en IA.

Se puede identificar un conjunto de consideraciones importantes que el sistema de motivación de un agente debe tener en cuenta. Aquí hay un conjunto de consideraciones concernientes a las unidades como las construcciones principales.

Activación proporcional. La activación de una unidad debe ser proporcional a las correspondientes compensaciones, o déficits, en aspectos relacionados (como alimentos o agua).

Oportunismo: Un agente necesita incorporar consideraciones concernientes a las oportunidades. Por ejemplo, la disponibilidad de agua puede llevar a preferir beber agua en lugar de recolectar alimentos (siempre que los déficits de alimentos no sean demasiado buenos)

Contigüidad de acciones: debe haber una tendencia a continuar con la secuencia de acción actual, en lugar de cambiar a una secuencia diferente, para evitar la sobrecarga de conmutación.

Persistencia: de manera similar, las acciones para satisfacer una unidad deben persistir más allá de la satisfacción mínima, es decir, más allá de un nivel de satisfacción apenas reducido para reducir la unidad más urgente a estar ligeramente por debajo de otras unidades.

Interrupción cuando sea necesario. Sin embargo, cuando surge un impulso más urgente (como "evitar el peligro"), las acciones para un accionamiento de prioridad más baja (como "dormir") pueden interrumpirse.

Combinación de preferencias: Las preferencias resultantes de diferentes unidades deben combinarse para generar una preferencia general algo mayor. Por lo tanto, se puede generar un candidato de compromiso que no sea el mejor para cualquier unidad, sino el mejor en términos de la preferencia combinada.

Un sistema bipartito de representación motivacional es el siguiente. Los objetivos explícitos de un agente pueden generarse en función de las estadísticas de la unidad interna del agente. Esta representación explícita de objetivos deriva de, y depende de, estados de impulso (implícitos). Específicamente, esto se refiere a las unidades primarias como las unidades que son esenciales para un agente y, para empezar, es muy probable que estén integradas (cableadas). Algunos ejemplos de unidades primarias de bajo nivel incluyen:

Obtenga comida, consiga agua y evite el peligro.

Estas unidades pueden implementarse en una red neuronal de retro propagación (preformada), que representa instintos evolutivos precableados.

Más allá de tales unidades de bajo nivel (con respecto a las necesidades fisiológicas), también hay unidades de mayor nivel. Algunos de ellos son primarios, en el sentido de estar "conectados". La "jerarquía de necesidades" de Maslow identifica algunos de estos impulsos. Algunas unidades de alto nivel particularmente relevantes incluyen: pertenencia, estima, autorrealización, etc.

Si bien las unidades primarias están integradas y son relativamente inalterables, también hay unidades "derivadas", que son secundarias, modificables y se adquieren principalmente en el proceso de satisfacer las unidades primarias. Las unidades derivadas pueden incluir: (1) unidades adquiridas gradualmente, mediante "condicionamiento"; (2) unidades configuradas externamente, a través de instrucciones dadas externamente. Por ejemplo, debido a la transferencia del deseo de complacer a los superiores en un deseo específico de cumplir con sus instrucciones, seguir las instrucciones se convierte en un impulso (derivado).

Se pueden establecer objetivos explícitos basados en estos impulsos [24]

4.2.3.4 - Subsistema meta-cognitivo

La meta-cognición se refiere al conocimiento sobre los propios procesos cognitivos y sus resultados. La meta-cognición también incluye el monitoreo activo y la consiguiente regulación y orquestación de estos procesos, generalmente al servicio de algún objetivo concreto. Esta noción de meta-cognición se operacionaliza dentro de CLARION.

En CLARION, el subsistema meta-cognitivo (MCS) está estrechamente relacionado con el subsistema motivacional. El MCS monitorea, controla y regula los procesos cognitivos en aras de mejorar el rendimiento cognitivo. El control y la regulación pueden consistir en establecer objetivos para el ACS, interrumpir y cambiar los procesos continuos en el ASC y el NACS, establecer parámetros esenciales del ACS y el NACS, etc. El control y la regulación también se llevan a cabo mediante el establecimiento de funciones de refuerzo para el ACS en función de los estados de accionamiento.

En este subsistema, hay muchos tipos de procesos meta-cognitivos disponibles para diferentes propósitos de control meta-cognitivo. Entre ellos, hay los siguientes tipos:

1. Puntería conductual
2. Filtros de información
3. Adquisición de información
4. Utilización de la información
5. Selección de resultados
6. Selección de modo cognitivo
7. Parámetros de configuración del ACS y el NACS

Estructuralmente, el MCS puede subdividirse en varios módulos. El nivel inferior consta de las siguientes redes (separadas): la red de establecimiento de objetivos, la red de función de refuerzo, la red de selección de entrada, la red de selección de salida, la red de configuración de parámetros (para establecer tasas de aprendizaje, temperaturas, etc.), etc. De manera similar, las reglas en el nivel superior (si existen) se pueden subdividir correspondientemente.

Este subsistema puede ser pre-entrenado antes de la simulación de cualquier tarea en particular (para capturar instintos evolutivos pre-cableados, o conocimiento / habilidades adquiridas de la experiencia previa) [25].

Con la separación de lo implícito y explícito en cada uno de los subsistemas una gran variedad de tipos de conocimientos se puede representar. Estos tipos de conocimiento no son solo importantes para modelar agentes individuales, sino que también son importantes para modelar interacciones sociales entre los agentes. Estos capturan las rutinas habituales del día a día copiándolas del mundo que involucra a otros agentes, elaboran planes para tareas específicas teniendo en cuenta otros agentes, contienen conocimiento explícito acerca del mundo y otros agentes, asociaciones implícitas creadas de experiencias anteriores para generar otros conocimientos que pueden involucrar otros agentes, etc. Modelos cognitivos de agentes serían menos capaces sin alguno de estos tipos de conocimientos. Simulaciones sociales serían además menos realistas sin ellos.

Por encima de ello, con la habilidad de aprender desde arriba hacia abajo y desde abajo hacia arriba, CLARION captura capacidades de aprendizaje más realistas de agentes

cognitivamente más realistas. La combinación de ambas direcciones de aprendizaje, especialmente desde abajo hacia arriba, posibilita el modelado de interacciones complejas de un agente y su ambiente en aprender múltiples tipos de conocimientos en muchas maneras diferentes. En particular esto posibilita la captura de aprendizajes socioculturales complejos junto a otras situaciones [26].

4.3 - Evaluación de criterios por arquitectura

Para que la comparación sea más visible, como se ha mencionado, se dará primero una calificación en una escala de 1 a 5 con su explicación correspondiente respecto a la rúbrica planteada, y luego, estas calificaciones serán resumidas en una tabla comparativa.

4.3.1 LIDA

Diseño de agente:

Desde su fundamento LIDA se subdivide en *codelets* y subsistemas pequeños que se encargan de tareas específicas. Luego en su dinámica de interacción en el ciclo cognitivo se genera el comportamiento. Estas porciones se encuentran bien determinadas y la implementación de cada una de ellas puede ser independiente y de tamaños razonables a la escala de agente que se pretende. Por esto tener un gran número de agentes sería factible con recursos razonables de hardware.

Existe un framework desarrollado para la implementación de LIDA, llamado LIDA framework, implementado en Java. Su última actualización fue realizada hace 5 años y su grado de utilidad actual podría verse comprometida a cambios, necesarios para ser utilizable con versiones actuales de Java. No obstante, está pensado para ser ejecutado en un entorno de escritorio corriente y no en grandes servidores. La escalabilidad de ejecutar múltiples instancias entonces no sería a priori un problema. Debido a su grado de desactualización del framework y poca actividad en la actualización del mismo podría pesar negativamente en su utilización, a pesar de la escalabilidad que le da el fundamento de su arquitectura. El puntaje es 4/5.

Ambiente:

El sensado de estímulos tanto internos como externos se encuentra completamente desacoplado de las demás partes, no habiendo un obstáculo desde su concepción que le impida relacionarse con ambientes de cualquier complejidad.

Del lado del framework se puede decir que como se encuentra desarrollado en Java la adaptabilidad, dado su soporte e implementaciones de librerías y baja complejidad al ser un lenguaje de uso común, se considera un punto fuerte para su uso. El puntaje es 4/5.

Percepción:

Si bien el sensado de estímulos se encuentra desacoplado, y existen un gran número de librerías para interactuar con distintos tipos de entradas y salidas dentro del sistema, la interacción concurrente de estos agrega una complejidad extra. A priori no sería algo sencillo de implementar con las bases del framework, aunque posible. Por ello se le da un puntaje de 3/5.

Control:

La arquitectura propone la toma de decisiones de acciones desde el espacio de trabajo basada en un mecanismo de competencia de la adaptabilidad del plan de acciones a la contribución del logro del objetivo. Estas decisiones podrían ampliarse para tener en cuenta no solo percepciones de adaptabilidad propia sino también ajena, comunicadas por otros agentes. Esto podría verse en la forma de sugerencia de plan de acción según un contexto por parte de un agente a otro, lo cual promueve positivamente el intercambio de decisiones en el control de acciones para un objetivo común. Esto es teóricamente prometedor, no obstante, desde la implementación del framework no existe estas capacidades desarrolladas. Para el intercambio de información entre procesos existen mecanismos maduros en el lenguaje para hacerlo. Por todo esto se le da un puntaje de 2/5.

Conocimiento:

El conocimiento y la memoria en LIDA son almacenadas explícitamente en redes de nodos interconectadas por lo que si múltiples agentes pueden acceder a las mismas redes o intercambiar información de estas redes, el conocimiento podría ser compartido. La única complejidad inherente en este ámbito será el múltiple acceso a redes junto a la actualización y mantenimiento de las mismas con lo cual se deberá definir un proceso de acceso y escritura. Por ello se le da un puntaje de 3/5.

Comunicación:

La comunicación, no solo a nivel de estímulos sino también de conocimiento, planes de acción y diferentes tipos de memorias, es totalmente transparente desde la modularidad de la arquitectura para los diferentes procesos encargados de utilizarlos. Por ello, el intercambio de estos elementos es a priori no sólo plausible sino también beneficioso en el intercambio directo. Luego, dada la vasta cantidad de librerías que existen para el intercambio de información de todo tipo en la plataforma donde se encuentra LIDA Framework, la comunicación entre procesos no sería un problema y sería viable. Por ello se le da un puntaje de 4/5.

4.3.2 CogPrime

Diseño de agente:

OpenCog es un framework en el cual la arquitectura CogPrime puede ser implementada. Es un framework monstruoso respecto a dimensiones y recursos necesarios para ser viable en su ejecución. Esta complejidad se la da su modelo de memoria y conocimiento e interacciones dentro del Atomspace. Además, múltiples partes dentro del framework no han sido aún implementadas. Esto haría que la implementación de múltiples agentes dentro de un mismo ambiente sea poco viable y además debe esperarse a su completitud para ser probada en sus interacciones. Debido a esto se le da un puntaje de 1/5.

Ambiente:

Tanto la arquitectura como el framework no tienen limitaciones claras en cuanto a la interacción con ambientes de alta complejidad más allá de los tipos de sensores y actuadores que podría necesitar. Sin embargo, la representación del ambiente en la forma de

conocimiento elegida es a la vez su ventaja y desventaja. Traducir elementos del ambiente en conocimiento es un proceso complejo, pero una vez esto es realizado interactuar con este conocimiento es muy potente. Añadiendo a su falta de madurez se le da un puntaje de 2/5.

Percepción:

La percepción del ambiente no solo por un agente sino por múltiples sería además compleja por los sensores necesarios y por los mecanismos de detección de patrones. Estos mecanismos pueden interferir entre agentes si se necesitan grandes recursos para su interpretación cuando varios agentes observen el espacio del ambiente. Por esto si bien no es intrínsecamente una limitante, su complejidad lo hace dificultoso para escalar a muchos agentes. Por ello se le da un puntaje de 2/5.

Control:

La comunicación entre agentes tendría una complejidad elevada debida a la forma en que el framework ejecuta las instancias de los subprocesos. No existe en principio el concepto de un todo donde pueda tomarse como límite de un agente para un conjunto de instancias de los subprocesos, menos aún de múltiples. Con lo cual el acuerdo entre agentes para la coordinación de tareas se muestra complejo en este escenario. Por ello se le da un puntaje de 1/5.

Conocimiento:

El conocimiento, representado en Atomspace en memoria difícilmente podrá ser accedido por múltiples instancias al mismo tiempo sin antes volcarse a un punto de interacción como archivos o conocimiento explícito legible por otro proceso. La comunicación a través de memoria entre procesos es compleja y puede que su implementación requiera cambios en el mismo framework. Debido a ello se le da un puntaje de 1/5.

Comunicación:

Comunicación entre agentes podría verse favorecida a través del intercambio de reconocimiento de patrones de acciones, dadas situaciones específicas del ambiente, atacando directamente a la incertidumbre de lo observado por un solo agente. No obstante, por el propio modelo de conocimiento, no es claro cuán posible es intercambiar decisiones, acciones o procedimientos específicos sin verse comprometido por la complejidad del manejo del hipergrafo. Por esto se le da un puntaje de 2/5.

4.3.3 CLARION

Diseño de agente:

Desde sus partes fundamentales CLARION cuenta con cuatro grandes subsistemas que interactúan entre ellos, donde como ya se mencionó cada uno tiene un propósito específico. En principio, por el propio diseño de esta intercomunicación de subsistemas y comunicación interna de diferentes procesos en cada uno de estos subsistemas, su

escalabilidad promete ser una gran ventaja dada la modularización de especialidades. Esto favorece la creación de un gran número de instancias basadas en un esquema común, con recursos razonables de hardware.

El framework, al igual que en LIDA se encuentra desarrollado en java, pero además existe una versión en C#. Su última versión oficial fue distribuida sin embargo en Octubre del año 2019, siendo sustancialmente más reciente que la versión de LIDA framework.

Por ello si bien fundamentalmente se encuentra más acoplado que LIDA respecto a su estructura arquitectónica, su soporte es más reciente en cuanto a implementación, y entonces se le da un puntaje de 4/5.

Ambiente:

El ACS, subsistema encargado de la interacción con el ambiente, es considerado el subsistema más importante dentro de CLARION el cual observa el mismo para guiar la decisión de toma de acciones, incluidas otras cualidades ya mencionadas. Esta observación en principio no se encuentra limitada a un tipo específico de ambiente ni tampoco limita que este ambiente sea completamente comprendido para la decisión de acciones. Por ello, debido a este enfoque donde explícitamente se lo considera parte fundamental para el ciclo de percepciones y decisiones se le da un puntaje de 5/5.

Percepción:

Si bien se mencionó que el subsistema ACS, parte fundamental de la arquitectura, es una clara ventaja para atacar el problema de la interacción con el ambiente, su complejidad promueve que la utilización masiva de los mismos recursos por parte de múltiples agentes pueda entrar en conflicto, y así limitar la cantidad máxima de agentes a interactuar sobre el mismo espacio dentro de un ambiente. Por ello se le da un puntaje de 3/5.

Control:

El control de acciones habiendo tomado decisiones en base a percepciones como así también a los objetivos y motivaciones propias del agente no es tan explícito como lo es en LIDA, y con ello la interrelación de acciones y conocimiento propio del agente es más acoplada. Por ello si bien podrían implementarse sugerencias de acciones entre agentes, la línea entre una acción explícita y una sugerencia dada un contexto no es clara. Por esto se da un puntaje de 2/5.

Conocimiento:

El conocimiento, desde su representación dual, implícita y explícita con sus interacciones de aprendizaje desde abajo hacia arriba y desde arriba hacia abajo, hace posible el intercambio entre agentes. No obstante, a medida que se adquiera más y más conocimiento esta interrelación entre conocimientos en cada agente se hará más compleja y así menos fácil de trasladar.

Además, las reglas definidas a partir del conocimiento implícito podrían ser similares, pero no iguales y favorecer a una mala interpretación de acciones a tomar por ser distintas en términos prácticos pero similares en concepto.

Por esto se le da un puntaje de 2/5.

Comunicación:

Desde su concepción la interacción de los subsistemas y los subprocesos dentro de ellos se realiza a través de la comunicación de mensajes, por lo cual sería una clara ventaja al intentar comunicaciones inter-agentes. No obstante, el contenido de esta comunicación podría ser difícil de ser interpretada si el conocimiento entre agentes es sustancialmente distinto. Por ello se le da un puntaje de 4/5.

4.4 - Tabla comparativa

Habiendo analizado cada uno de los elementos de la comparación pretendida en cada una de las arquitecturas, se resume en la siguiente tabla los puntajes obtenidos para luego ser utilizada como base en la conclusión final en la siguiente sección.

Criterio / Arquitectura	LIDA	CogPrime	CLARION
Diseño de agente	4	1	4
Ambiente	4	2	5
Percepción	3	2	3
Control	2	1	2
Conocimiento	3	1	2
Comunicación	4	2	4
Promedio	3,3’	1,5	3.3’

5 - Conclusiones y trabajo futuro

Luego de este análisis se pueden destacar ciertos elementos importantes que no son solo interpretaciones técnicas de las arquitecturas sino también reflexiones para trabajo futuro.

La novedad de la utilización de agentes basados en arquitecturas de inteligencia artificial general para generar o simular comportamientos en sistemas multi-agentes radica en la importancia de dotar a los agentes con una percepción más compleja y racional del ambiente, sus interacciones y los demás agentes. Esto acerca más a la realidad en el caso de la simulación de comportamientos humanos, donde la racionalidad es un aspecto fundamental de la toma de decisiones, aunque no el único.

No obstante, la implementación de un sistema multi-agentes con estas características agrega una enorme complejidad y dificultad, por el simple hecho de la implementación de cada agente y las herramientas disponibles en cada arquitectura. Por ello, durante el análisis fue muy relevante el framework implementado que soporta la arquitectura.

Si se enfoca directamente en el resultado numérico de la comparación se ve que tanto LIDA como CLARION obtuvieron el mismo puntaje total, pero con puntajes diferentes en los distintos criterios. Con esto puede notarse que una buena decisión sobre la arquitectura elegida dependerá del tipo de sistema multi-agentes que se quiera desarrollar. Si el sistema multi-agentes se encuentra mayormente inclinado hacia un aspecto donde se ha destacado la arquitectura en este análisis sería razonable elegirla.

Además, existen múltiples otros criterios posibles que van más allá de este análisis. Un ejemplo de ello es el análisis de las emociones. CLARION tiene una solución más explícita respecto a la influencia de las motivaciones o sentimientos en las decisiones tomadas respecto a LIDA. Con lo cual, si en este sistema multi-agentes deseado, las emociones o motivaciones juegan un rol muy importante debería de ser elegida CLARION por sobre LIDA.

Por otra parte, la delegación de acciones explícitas en LIDA es mucho más clara, no solo desde la comunicación sino también desde la representación del conocimiento. Por ello, si el sistema multi-agentes deseado declina su importancia sobre la coordinación de acciones a través de comunicación explícita de estas mismas, debería de elegirse LIDA por sobre CLARION.

Para realizar una comparación más detallista en términos de performance e implementaciones se necesita un análisis más profundo aplicado a un contexto de sistema multi-agentes particular, y no general. Con el cual probablemente se necesiten implementaciones parciales para tener datos prácticos para medir esta performance, más allá desde el punto de vista teórico.

A futuro, se pretende tomar una de las arquitecturas y realizar una implementación particular para la simulación de un sistema multi-agentes y comprobar o no sus fortalezas y debilidades analizadas. Estos sistemas multi-agentes deseados se encuentran inclinados, por una apreciación personal, a tener en cuenta aspectos emocionales y motivacionales en la toma de decisiones y aprendizaje relacionados a la personalidad. Por ello se elegirá CLARION como la arquitectura base para el desarrollo de sistemas multi-agentes, para la simulación de comportamientos emergentes influenciados por motivaciones y emociones específicas relacionados a la personalidad.

6 - Bibliografía

1 - Vijay Kanade, What Is General Artificial Intelligence (AI)? 2022 Sitio web: https://www.toolbox.com/tech/artificial-intelligence/articles/what-is-general-ai/

2 - Stuart Russell, Peter Norvig. (2009). Artificial Intelligence - A Modern Approach (3rd edition) (ISBN 13: 978-0-13-604259-4 ISBN-10: 0-13-604259-7). New Jersey: Prentice Hall. 1-5

3 - Jeff Kerns, What's the Difference Between Weak and Strong AI? 2017 Sitio web: http://www.machinedesign.com/robotics/what-s-difference-between-weak-and-strong-ai

4 - Dmitriy Smuschenko, Artificial Intelligence: Weak AI vs. Strong AI. 2016, de - Sitio web: https://vironit.com/artificial-intelligence-weak-ai-vs-strong-ai

5 - Pei Wan, Ben Goertzel. (2012). Deep Reinforcement Learning as Foundation for Artificial General Intelligence. En Theoretical Foundations of Artificial General Intelligence (89 - 91). Paris, France: Atlantis Press.

6 - Pei Wang, Ben Goertzel. (2012). A New Constructivist AI: From Manual Methods to Self-Constructive System. En Theoretical Foundations of Artificial General Intelligence(148-150). Paris, France: Atlantis Press

7 - Javier Snaider, Ryan McCall, Stan Franklin. (2012). The LIDA Framework as a General Tool for AGI.: Cognitive Computing Research Group

8 - SOAR. SOAR HOME. 2018, de - Sitio web: https://soar.eecs.umich.edu

9 - OpenCog Foundation. Build better minds together. 2018, de - Sitio web: https://opencog.org/

10 - Alexei Samsonovich. COMPARATIVE TABLE OF COGNITIVE ARCHITECTURES. 2012, de - Sitio web: http://bicasociety.org/cogarch/architectures.php

11 - A. M. Turing. (1950). COMPUTING MACHINERY AND INTELLIGENCE. En MIND(433-460). United Kingdom

12 - Blaize Zarega. (2017). The Turing test is tired. It's time for AI to move on. 2017, de - Sitio web: https://www.huffingtonpost.com/entry/the-turing-test-is-tired-its-time-for-ai-to-move_us_59bfae52e4b02c642e4a187a

13 - Pei Wang, Ben Goertzel. (2012). TheArchitectureofHuman-LikeGeneral Intelligence. En Theoretical Foundations of Artificial General Intelligence(142). Paris, France: Atlantis Press.

14 - Pei Wang, Ben Goertzel. (2012). TheArchitectureofHuman-LikeGeneral Intelligence. En Theoretical Foundations of Artificial General Intelligence(144). Paris, France: Atlantis Press.

15 - Let's think about the applications of strong AI. 2012, de - Sitio web: https://www.reddit.com/r/artificial/comments/vp1a6/lets_think_about_the_applications_of_strong_ai/

16 - Steve Strain, Stan Franklin. (2017). Cognitive Computing Research Group. 2017, de Cognitive Computing Research Group Sitio web:

17 - Computational LIDA Group . (2011). LIDA Framework Software NonExclusive, Non-Commercial Use License. 2011, de Cognitive Computing Research Group Sitio web: http://ccrg.cs.memphis.edu/assets/papers/2010/LIDA-framework-non-commercial-v1.0.pdf

18 - Ben Goertzel. (October 2, 2012). CogPrime: An Integrative Architecture for Embodied Articial General Intelligence.

19 - Ron Sun. (October 11, 2004). The CLARION Cognitive Architecture: Extending Cognitve Modeling to Social Simulation

20 - Ron Sun. (October 11, 2004). The CLARION Cognitive Architecture: Extending Cognitve Modeling to Social Simulation [pag1]

21 - Ron Sun. (October 11, 2004). The CLARION Cognitive Architecture: Extending Cognitve Modeling to Social Simulation [pag5]

22 - Ron Sun. (October 11, 2004). The CLARION Cognitive Architecture: Extending Cognitve Modeling to Social Simulation [pag7]

23 - Ron Sun. (October 11, 2004). The CLARION Cognitive Architecture: Extending Cognitve Modeling to Social Simulation [pag12]

24 - Ron Sun. (October 11, 2004). The CLARION Cognitive Architecture: Extending Cognitve Modeling to Social Simulation [pag14]

25 - Ron Sun. (October 11, 2004). The CLARION Cognitive Architecture: Extending Cognitve Modeling to Social Simulation [pag17]

26 - Ron Sun. (October 11, 2004). The CLARION Cognitive Architecture: Extending Cognitve Modeling to Social Simulation [pag20]

27 - Wiley - Wooldrigde An Introduction to Multi Agent Systems [pag19]

28 - Vicente J. Botti Navarro Adriana Giret Boggino. (2013). Aplicaciones Industriales de los Sistemas Multiagente. Departamento de Sistemas Informáticos y Computación Universidad Politécnica de Valencia: [pag 7, 8]

29 - Nikos Vlassis. (2003). A Concise Introduction to Multiagent Systems and Distributed AI. Universiteit van Amsterdam [pag 7]

30 - Nikos Vlassis. (2003). A Concise Introduction to Multiagent Systems and Distributed AI. Universiteit van Amsterdam [pag 1-4]

31 - Ron Sun. (2005). Cognition And Multi-Agent Interaction From Modeling to social simulation. Rensselaer Polytechnic Institute: Cambridge University Press.

32 - Stan Franklin, Uma Ramamurthy, Sidney K. D'Mello, Lee McCauley, Aregahegn Negatu, Rodrigo Silva L., Vivek Datla. (2007). LIDA: A Computational Model of Global Workspace Theory and. Institute for Intelligent Systems and the Department of Computer Science, The University of Memphis, Memphis, TN 38152, USA: Association for the Advancement of Artificial Intelligence.

33 - David Friedlander, Stan Franklin. (2014). LIDA and a Theory of Mind. 2014, de Cognitive Computing Research Group Sitio web: http://ccrg.cs.memphis.edu/assets/papers/LIDA%20and%20a%20Theory%20of%20Mind%20v20.pdf