Análisis, diseño e implementación de un agente deliberativo para extraer contextos definitorios en textos especializados*

María Mercedes Suárez de la Torre**
Luis Fernando Castillo Ossa***
Carmenza Ríos Cardona****
Germán Mauricio Muñoz*****
Jorge Aranzazu Álvarez******

Resumen

Este artículo presenta los resultados de la primera fase del proyecto en curso: Sistema Multiagente para la extracción automática de contextos definitorios basado en ontologías para la Web semántica. El objetivo principal de este artículo es mostrar el análisis, diseño e implementación de un agente deliberativo, con un mecanismo de aprendizaje supervisado, que permite identificar contextos definitorios en textos especializados. Con el fin de lograr dicho objetivo, se ha seleccionado, para la aplicación del agente, la metodología GAIA que proporciona un conjunto de pasos incrementales, mediante la construcción de sistemas basados en agentes como un proceso de diseño organizacional. La extracción de contextos definitorios se ha realizado utilizando el conjunto de herramientas GATE, que permite reconocer expresiones regulares en documentos marcados sintáctica y semánticamente. Como resultado de la interacción entre el agente deliberativo y el corpus de textos seleccionado, se obtienen los contextos definitorios de manera semiautomática. Dicha extracción semiautomática funciona mediante del diseño de las siguientes interfaces: interface básica de aplicación, Interface en interacción con la plataforma JADE e interface de comunicación entre agentes. El diseño y la puesta en marcha de cada una de estas interfaces, ha permitido concluir que los trabajos realizados en cuanto a la extracción de contextos definitorios y patrones lingüísticos, no brindan información suficiente desde el punto de vista sintáctico y semántico, para que la máquina pueda reconocerlos y realizar una búsqueda y recuperación más refinada, utilizando un número mínimo de fases.

* Artículo derivado de la investigación «*Sistema Multiagente para la extracción automática de contextos definitorios basado en ontologías para la Web semántica»,* financiado por COLCIENCIAS y la Universidad Autónoma de Manizales (código 1219-405-20249). Este proyecto está en ejecución y es desarrollado por los grupos de investigación CITERM e Ingeniería de Software de la Universidad Autónoma de Manizales, Manizales, Colombia. El proyecto inició en el año 2008 y finaliza en marzo de 2010.

** Investigadora principal. Doctora en Lingüística Aplicada, Grupo de Investigación CITERM. Profesora titular, Instituto de idiomas, Universidad Autónoma de Manizales. Manizales, Colombia. mercedessuarez@autonoma.edu.co.

*** Investigador principal. Doctor en Informática, Grupo de investigación en Ingeniería de Software. Profesor Asociado, Departamento de Ciencias Computacionales, Universidad Autónoma de Manizales. Manizales, Colombia. lfcastil@autonoma.edu.co.

**** Coinvestigadora. Magíster en Didáctica del inglés, Grupo de Investigación CITERM. Profesora instructora, Instituto de Idiomas, Universidad Autónoma de Manizales. Manizales, Colombia. carmrios@autonoma.edu.co.

***** Asistente de investigación. Ingeniero de Sistemas, Universidad Autónoma de Manizales. Manizales, Colombia. gmauricio.munoz@gmail.com.

****** Asistente de investigación. Ingeniero de Sistemas, Universidad Autónoma de Manizales. Manizales, Colombia. jorgearanzazu@hotmail.com.

Palabras clave: Agente deliberativo, contextos definitorios, textos especializados, patrones lingüísticos, extracción de información semiautomática.

Cómo citar este artículo: SUÁREZ DE LA TORRE, María Mercedes, *et al*. Análisis, diseño e implementación de un agente deliberativo para extraer contextos definitorios en textos especializados. *Revista Interamericana de Bibliotecología*. Jul.– Dic. 2009,vol 32, no. 2; p. 5-84.

Artículo recibido: 14 de septiembre de 2009. Aprobado: 24 de noviembre de 2009.

Abstract

This paper presents the results of the first stage comprised in the research project entitled: *Multi-agent System for Defining Contexts Extraction, based on ontologies for the Semantic Web*. This article mainly aims at showing the analysis, design and implementation of a deliberative agent with a supervised learning mechanism, which permits the identification of defining contexts in specialized texts. The GAIA methodology has been selected to apply the agent in order to reach the stated goal. This methodology provides an increasing set of steps, based on agent- based systems as an organizational design process.

The process of defining context extraction has been carried out by means of the GATE tool, which allows the detection of regular expressions in documents syntactically and semantically marked. The defining contexts are obtained in a semi-automatic way as a result of the interaction between the deliberative agent and the corpus selected. The semi-automatic extraction works by means of the design of the following interphases: basic application interphase, interacting with the JADE platform interphase, and agent communication interphase. After the design and the implementation of each of the inter phases mentioned above, it has been concluded that the research works dealing with the defining context extraction and linguistic patterns, do not provide the necessary information, from the syntactic and semantic aspect, to make the machine recognize the defining contexts and to develop a more refined search and retrieval, while using a minimum phase amount.

Keywords: Deliberative agent, defining contexts, specialized contexts, linguistic patterns, semi-automatic information extraction.

How to cite this article: SUÁREZ DE LA TORRE, María Mercedes, et al. Analysis, design and implementation of a deliberative agent for defining context extraction in specialized texts. *Revista Interamericana de Bibliotecología*. Jul.– Dic. 2009, vol 32, n° 2; p. 59-84.

Introducción

El trabajo interdisciplinario e inter-grupal ha cobrado importancia en campos de conocimiento como la terminología, la traducción y la documentación, entre otros y ha permitido un avance significativo en lo concerniente al uso de tecnologías, específicamente en el tratamiento de corpus especializados para extraer o recuperar información de manera automática. En los últimos años, los estudios más aplicados de la terminología han intentado incursionar en el reconocimiento y la extracción de términos y unidades mayores (combinaciones léxicas) y, para ello, se han tenido en cuenta las descripciones desde el punto de vista lingüístico.

Dentro de este marco, los grupos de investigación CITERM e Ingeniería de Software, de la UAM, realizan un proyecto para poner en marcha un sistema multiagente basado en ontologías, que permita extraer contextos definitorios para la Web semántica[1]. En este sentido, en primer lugar, desarrollamos con la ayuda de agentes, un proceso computacional que recurre a la autonomía y a la funcionalidad de comunicación de una aplicación y, en segundo lugar, aplicamos una serie de técnicas como: sistemas de gestión de conocimiento e Inteligencia Artificial Distribuida (IAD), con el fin de solucionar problemas, mediante la interacción entre diferentes agentes, de tal modo que juntos permitan alcanzar la funcionalidad deseada. Dicha funcionalidad está orientada a la extracción automática de contextos definitorios en la Web y, para ello, consideramos pertinente trabajar con ontologías de dominio específico.

En este artículo presentamos los resultados obtenidos en la primera fase del proyecto, cuyo objetivo es el análisis, diseño e implementación de un agente deliberativo, con un mecanismo de aprendizaje supervisado que permite identificar contextos definitorios en textos especializados. Con el fin de lograr dicho objetivo, hemos seleccionado para la aplicación del agente la metodología GAIA, que proporciona un conjunto de pasos incrementales, porque el aspecto trascendental es la construcción de sistemas basados en agentes como un proceso de diseño organizacional.

El desarrollo de esta metodología sigue las indicaciones de Wooldrige *et al*. [1] y Wei Huang et al. [2]. Los agentes deliberativos están conformados por *deseos*, *creencias* e *intenciones* (Raja & Lesser [3]). Las creencias están representadas por patrones lingüísticos descritos sintácticamente, los cuales son aplicados a un corpus; los deseos se entienden como la finalidad del agente para identificar los contextos definitorios, y las intenciones hacen referencia a una etapa de aprendizaje supervisado por un experto, de modo que el agente pueda identificar de manera automática dichos contextos. Para la extracción de contextos definitorios hemos utilizado el conjunto de herramientas GATE[2], el cual contiene un sistema de extracción de información denominado ANNIE (*A Nearly New Information Extraction System*), que trabaja con base en algoritmos de estados finitos y el lenguaje JAPE (*Java Annotations Patterns Engine*), el cual, a su vez, permite reconocer expresiones regulares en documentos marcados sintáctica y semánticamente.

1. Este proyecto fue aprobado por el Consejo del Programa Nacional de Desarrollo Tecnológico Industrial y Calidad, correspondiente a la línea de recuperación contingente de COLCIENCIAS (Instituto Colombiano para el Desarrollo de la Ciencia y la Tecnología), identificado con código 1219-405-20249.

2. GATE. *General Architecture for Text Engineering.* [En línea]. Disponible en: http://gate.ac.uk/ [Consulta: 10 de septiembre de 2009]

1. Referente conceptual

1.1. Sistema multiagente y agentes deliberativos: fundamentos de los agentes

Aunque la comunidad científica que se dedica a investigar sobre agentes es muy amplia, existen dos campos sobre los cuales ha tenido mayor influencia: la Inteligencia Artificial Distribuida (IAD) y la Computación Distribuida (CD).

Antes de empezar a tratar el concepto de IAD, es conveniente introducir el término Inteligencia Artificial (IA), que surge en el año 1956, durante una conferencia científica convocada por McCarthy, a la cual asistieron Minsky, Newell & Simon, entre otros. A partir de esta conferencia, dichos investigadores propusieron modelar la inteligencia humana en sistemas computacionales.

Iglesias [4] define la Inteligencia Artificial Distribuida (IAD) como «aquella parte de la IA que se centra en comportamientos inteligentes colectivos, productos de la cooperación de diversos agentes». El concepto de agente como entidad computacional aislada evoluciona desde la IAD debido al influjo de la Ingeniería del Software. Actualmente se considera la Programación Orientada al Agente, desde el punto de vista de la Ingeniería del Software, como la metodología capaz de superar las limitaciones de la Programación Orientada a Objetos.

Los agentes de software han evolucionado a partir de los Sistemas Multiagente (SMA), Pa Pa and Ni Lar [5]; Shafiq et al [6]; Changjian and Yao [7]; y están basados en tres áreas: Inteligencia Artificial Distribuida (IAD), Shaw et al. [8], Resolución de Problemas Distribuidos (RPD) e Inteligencia Artificial Paralela (IAP). En este sentido, los sistemas multiagente heredan las motivaciones, objetivos y potenciales beneficios de la IAD. Por ejemplo, gracias a la computación distribuida, los agentes de software heredan las características de modularidad, velocidad y confiabilidad. Igualmente gracias a la IA, se heredaron características de manipulación del conocimiento, facilidad de mantenimiento, reusabilidad e independencia de la plataforma, Huhns & Singh [9].

Tal vez la capacidad más novedosa aportada por la Inteligencia Artificial Distribuida a la teoría de agentes es el concepto de *inteligencia,* Mitkas et al [10]. Este concepto está íntimamente relacionado con la Inteligencia Artificial, tanto en los métodos de representación del conocimiento, como en la potencia de los algoritmos de razonamiento.

1.2. Agentes y estructura

Algunos investigadores definen agente como un sistema computacional que, además de las características de autonomía, sociabilidad y reactividad, se comprende

mediante conceptos aplicados usualmente a los humanos. Todavía no existe un consenso sobre una definición formal aceptada por todos los investigadores de este campo. Por ejemplo, es común que en IA un agente se caracterice por la utilización de nociones mentales tales como conocimiento, creencias, intenciones y obligaciones Shoham [11]; otros consideran a los agentes con emociones Bates *et al* [12]; Bates [13]. Otros dan a los agentes rasgos propios de los humanos, como una representación visual por medio de un icono o animación Maes [14]. En la actualidad algunos autores como Acay et al [15], desean extender las capacidades de los agentes tomando como referencia la noción utilidades-herramientas de software.

Aunque muchos autores han destacado que el concepto *agente* tiene una naturaleza muy genérica y adaptativa, es necesario caracterizarlo y, para ello, hemos asumido la definición estipulada por FIPA [16].

> «Un agente es un proceso computacional que implementa la autonomía y la funcionalidad de comunicación de una aplicación». (FIPA SC00023J).

Otra definición del concepto *agente* se encuentra en Tecuci [17]:

> «Un **agente inteligente** es un sistema basado en conocimiento que percibe su entorno (el cual puede ser el mundo físico, un usuario a través de una interfaz gráfica de usuario, un grupo de otros agentes, la internet, u otros ambientes complejos); razonan para interpretar percepciones, infieren, resuelven problemas y definen acciones; actúan sobre el entorno para materializar un conjunto de objetivos o tareas para las cuales fue diseñado. El agente interactúa (Ver **Figura 1**) con un humano o algún otro agente a través de algún tipo de lenguaje de comunicación de agentes y no obedece ciegamente, pero puede tener la habilidad de modificar requerimientos, responder preguntas de aclaración, o incluso rehusarse a satisfacer ciertas peticiones. Un agente puede aceptar solicitudes de alto nivel de acuerdo con lo que quiera el usuario, y puede decidir cómo satisfacer cada solicitud con algún grado de independencia o autonomía, exhibiendo comportamiento orientado a objetivos y escogiendo dinámicamente qué acciones llevar a cabo, así como en qué secuencia. Puede colaborar con el usuario para mejorar el cumplimiento de sus tareas o puede tomar estas tareas en beneficio del usuario, y para hacer esto emplea algún conocimiento o representación de los objetivos o deseos del usuario. Puede monitorear eventos o procedimientos para el usuario, puede aconsejar al usuario sobre cómo ejecutar una tarea, puede entrenar o enseñar al usuario, o puede ayudar a diferentes usuarios».

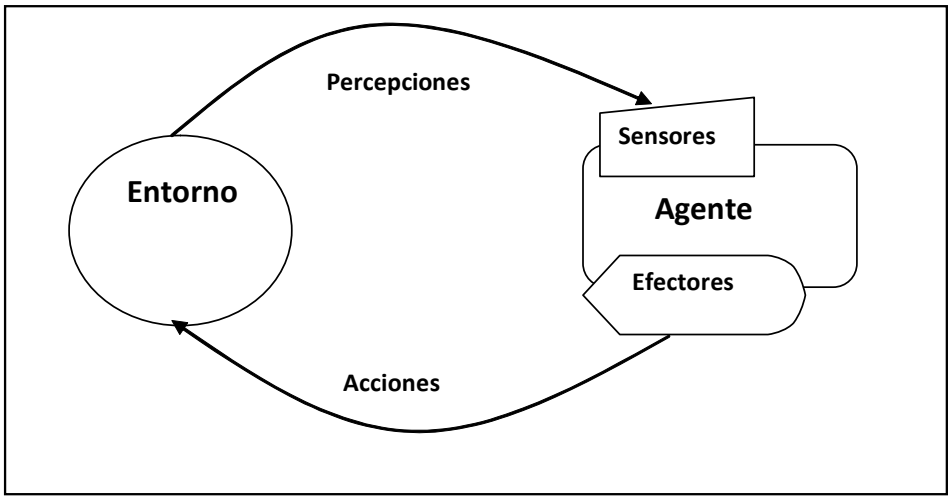

Figura 1. Interacción agente – entorno. Fuente: Rusell y Norvig [18]

1.2.1. Arquitectura de agentes

Una arquitectura define los mecanismos que permiten interconectar los componentes, tanto de software como de hardware Changjian and Yao [19]. En los agentes, las arquitecturas son las relaciones que fluyen entre las entradas (sensores), las salidas (actuadores) y el razonamiento interno del agente.

El programa del agente establece la correspondencia entre ciertas percepciones y ciertas acciones. Pero, una vez definido, el programa se ejecuta en algún tipo de computador con sensores físicos y actuadores. Este soporte para el programa se conoce como arquitectura y, en general, permite que las percepciones de los sensores estén disponibles para el programa y que los actuadores pongan en marcha las acciones generadas.

Una arquitectura de agentes describe la interconexión entre los diferentes módulos que constituyen el agente. Existen, al menos, dos usos del término arquitectura:

- Arquitectura como estructura general, refiriéndose a una abstracción que es común a muchos ejemplos de diseños concretos.
- Arquitectura como implementación concreta, refiriéndose a una de esas instancias de tales diseños.

En este trabajo, el término arquitectura se usa con base en el primer sentido; es decir, una arquitectura es una colección de características comunes a una clase de entidades. Cada ejemplo de arquitectura está formado de subestructuras que coexisten e interactúan con varias capacidades y roles funcionales. Una subestructura puede ser de nuevo una arquitectura. La arquitectura de un sistema complejo puede explicar cómo sus capacidades y comportamientos sobrepasan las capacidades, los comportamientos y las relaciones de sus componentes. En los agentes, las arquitecturas correspondientes deben permitir la implementación de las diferentes características recogidas en la teoría que los define; no obstante, dado que una arquitectura define los tipos de módulos de procesamiento de información, debe aparecer la interconexión existente entre ellos en un modelo adecuado de agente.

1.2.1.1. Arquitecturas deliberativas

Estas arquitecturas se caracterizan por la utilización de modelos de representación simbólica del conocimiento Gandon [20]; suelen basarse en la teoría clásica de planificación, en la que se parte de un estado inicial. Existe un conjunto de planes y un estado objetivo del cual se parte. Es muy generalizada la idea de diseñar, en estos agentes, un sistema de planificación que permita determinar el conjunto de pasos que van de un estado inicial a un estado final u objetivo. En estas arquitecturas, las decisiones pueden tomarse con base en mecanismos de razonamiento ejecutados mediante diferentes estrategias.

Según Corchado y Molina [21], cuando se decide implantar una arquitectura deliberativa, hay que buscar, en primer lugar, una descripción simbólica adecuada del problema, e integrarla en el agente para que éste pueda razonar y llevar a cabo las tareas encomendadas en el tiempo preestablecido. Aunque parece una cuestión trivial, debido a la complejidad de los algoritmos de manipulación simbólica, es un aspecto que requiere gran atención, especialmente si se tiene en cuenta que los agentes se desenvuelven en dominios específicos, en los que tienen que responder a estímulos en tiempo real. Dentro de estas arquitecturas, cabe destacar aquellas que basan su realización en el modelo BDI (*Belief*, *Desire*, *Intention*). Éste es uno de los modelos más utilizados hoy en día, Rao *et al* [22]. En la **Figura 2** se representa la arquitectura BDI:

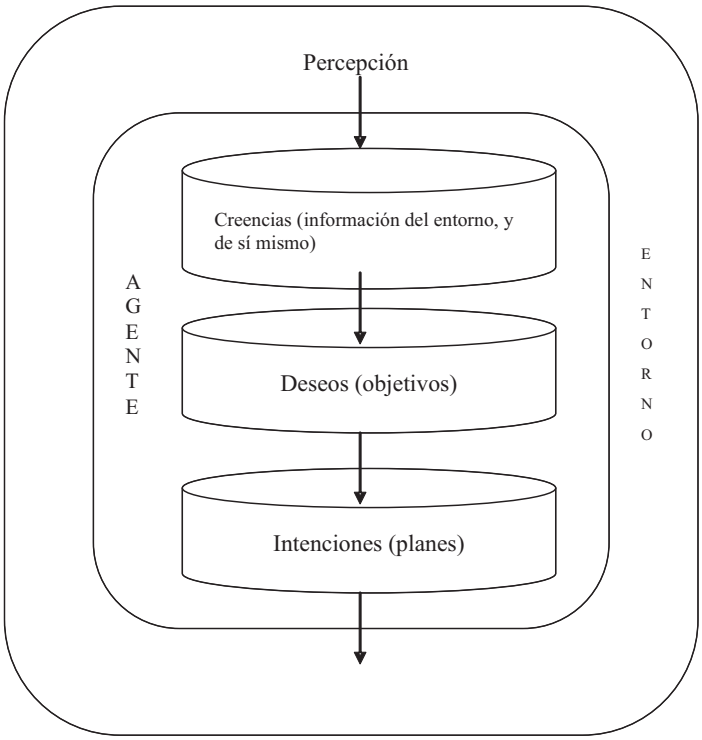

Figura 2. Agentes deliberativos

2. Contextos y patrones definitorios

Según Alarcón y Sierra [23], «una necesidad común en el trabajo terminológico es la identificación de información relevante sobre términos en textos especializados». Actualmente existe un creciente interés por la automatización en la búsqueda de ocurrencias de patrones definitorios, apoyada en el marcaje morfosintáctico de corpus especializados. En este mismo trabajo, Alarcón y Sierra afirman que cuando el autor de un texto especializado define un término, lo hace mediante contextos definitorios, en los cuales se utiliza una serie de patrones que pueden ser reconocidos automáticamente. Desde esta perspectiva, y apoyados en los avances de las investigaciones realizadas por el grupo de Ingeniería Lingüística de la UNAM, en este trabajo entenderemos *contexto definitorio* como:

> «Todo aquel fragmento textual de un documento especializado donde se aporta información útil para definir un término. Estas unidades están formadas por un término (**T**) y una definición (**D**), los cuales se encuentran conectados mediante patrones definitorios (**PD**)». Alarcón y Sierra [23].

Existen investigaciones que tratan el problema de la extracción automática de contextos definitorios desde una perspectiva teórico-descriptiva. Pearson [24] describe el comportamiento de los términos en el contexto de aparición y señala que cuando un autor define un término, suele recurrir a patrones tipográficos, para resaltar visualmente la presencia del término o la definición, y a patrones léxicos y metalingüísticos, para ligar los dos elementos anteriores mediante estructuras sintácticas, Alarcón y Sierra [23].

Meyer [25] sostiene que, en un texto especializado, los patrones definitorios que conectan los términos con su definición pueden también introducir claves que permitan reconocer automáticamente el tipo de definición presente en los contextos definitorios, así como elaborar automáticamente una red conceptual.

En Alarcón y Sierra [23], observamos que existen, también, investigaciones aplicadas que han partido de los estudios teórico-descriptivos, con el fin de elaborar metodologías para la extracción automática de contextos definitorios; estudios específicos para el reconocimiento automático de definiciones en textos médicos, Klavans & Muresan [26]; estudios sobre la identificación automática de definiciones, para sistemas de pregunta-respuesta, Saggion [27]; investigaciones para la extracción automática de información metalingüística, para terminología, Rodríguez [28] y estudios relacionados con la elaboración automática de ontologías, Malaisé *et al*. [29].

2.1.Patrones definitorios

Numerosas variantes denominativas, tanto en inglés como en castellano, acompañan el término patrón lingüístico, tal como lo refieren Marshman et al. [30]: *formulae* (Lyons, 1977) , *diagnostic frames* o *test frames* (Cruse, 1986), *frames* (Winston *et al*., 1987), *knowledge probes* (Ahmad y Fulford, 1992), *definitional metalanguage*; y *defining expositives* (Pearson, 1998), *speech patterns* o *patterns* (Meyer, 2001), *Operadores Metalingüísticos Explícitos* (OME) (Rodríguez, 1999), *patrones* (Faber, 2001), marcadores *lingüísticos* (Bowker y L'Homme, 2004), *marcadores de reformulación* (*MR*) (Bach, 2005).

Toda esta variedad de denominaciones se refiere a un mismo concepto que Condamines [31] define como:

> «A discursive structure used as an indication of the possible transition from the discourse to a model, allowing the more or less direct construction of a model in the form of a semantic relation depending on its relation with the context».

Observamos que la definición de Condamines se basa en una perspectiva semántica, al tratar de definir los patrones como una estructura discursiva. Esta definición, aunque útil en la aplicación de nuestro trabajo, se adecua de manera

más precisa en una etapa posterior de nuestro proyecto, cuando abordaremos el tema de las relaciones conceptuales y ontológicas.

Consideramos que para los fines de este artículo, la definición de Marshman et al. [30] es bastante adecuada:

> «Words, word combinations or paralinguistics features of texts which frequently indicate conceptual relations».

En este trabajo tendremos en cuenta los patrones definitorios, es decir, aquellas secuencias léxico-sintácticas que permiten tanto a los terminólogos, como a los informáticos estudiar los términos y sus definiciones en detalle.

Los patrones lingüísticos de diversa índole han sido clasificados (Ver **Tabla 1**):

Marshman *et al.* [30]	**Alarcón y Sierra** [32]
Patrones léxicos. Son los más visibles y consisten en palabras o grupos de palabras que indican las relaciones conceptuales.	**Patrón tipográfico o paralingüístico.** Secuencia léxica caracterizada por la presencia de elementos textuales o paralingüísticos como marcas de puntuación (dos puntos, coma, negrita, cursiva).
Patrones gramaticales. Implican combinaciones de estructuras grama–ticales que ofrecen relaciones semánticas entre conceptos, aunque de un modo más limitado que los patrones léxicos.	**Patrones sintácticos.** Secuencia encontrada en un contexto definitorio que incluye una predicación verbal o predicación pragmática.
Patrones paralingüísticos. Se trata de los elementos textuales, como por ejemplo, comas, paréntesis, puntos, etc.	**Patrones mixtos.** Aspectos léxicos y aspectos tipográficos. Tienen una estructura más sólida y resaltan visual y gramaticalmente el contexto definitorio.
	Patrones compuestos. Secuencias léxicas identificadas en un contexto definitorio, en el cual se definen dos términos distintos.

Tabla 1. Clasificación de patrones lingüísticos

Hemos observado que estas clasificaciones, aunque útiles desde la perspectiva de la descripción lingüística, no brindan información detallada y desglosada, de tal modo que la máquina pueda reconocer los patrones al nivel propuesto. Dicho de otro modo, la máquina requiere, tal y como veremos más adelante, pasar por varias fases que permitan refinar la búsqueda, sea de manera secuencial o paralela.

3. Metodología

En este apartado explicaremos la metodología utilizada en la primera fase de este proyecto y la describiremos en dos fases:

- Metodología para la extracción de los contextos definitorios.
- Metodología para la construcción del agente deliberativo.

3.1. Metodología para la extracción de los contextos definitorios

3.1.1. Constitución del corpus

Con el fin de realizar la extracción automática de contextos definitorios, seleccionamos un corpus, en inglés, de enfermedades neurológicas, conformado por 39 textos que contienen 170.000 palabras. Los textos seleccionados representan un mismo grado de especialidad; son textos del ámbito científico-académico y corresponden a la comunicación entre pares.

3.1.2. Descripción del conjunto de herramientas conocido como GATE[3]

Para extraer los contextos definitorios de manera automática se seleccionó *GATE* (*General Architecture for Text Engineering*); para ello, tuvimos en cuenta los siguientes criterios:

- Es una herramienta de libre acceso.
- Es una herramienta orientada al procesamiento de lenguaje natural en varios idiomas.
- Presenta una gran versatilidad en los sistemas que la conforman.

3.1.2.1. Herramienta *GATE*

A nuestro modo de entender, la sigla *GATE* tiene una doble connotación: por un lado, al referirse a una arquitectura para la ingeniería de textos, se quiere significar que hay una reorganización, re-codificación o reconfiguración de los contenidos lingüísticos del texto, de tal manera que puedan ser analizados mediante herramientas de ingeniería de sistemas informáticos; por otro lado, *GATE*, como palabra inglesa y no sólo como sigla, equivale a la palabra

3. Tras revisar cuatro programas informáticos para el análisis lingüístico de textos, seleccionamos GATE porque ofrece un mayor rendimiento en cuanto a velocidad, segmentación de símbolos, etiquetaje gramatical, análisis superficial de frases, visualización de etiquetas y la posibilidad de marcaje en diferentes idiomas. Los otros programas considerados fueron: *Machinese phrase tagger*, *Freeling* y *NLprocessor*.

castellana *puerta*, lo que significa que este programa se constituye en una entrada analítica que permite pasar de un texto plano a un texto etiquetado sintácticamente. Este paquete informático consta de un conjunto de herramientas desarrollado en lenguaje Java, diseñado en la Universidad de Sheffield (Inglaterra). Ha sido utilizado mundialmente por científicos, compañías, profesores y estudiantes, para todo tipo de tareas en el procesamiento de lenguaje natural (PLN); la herramienta incluye la extracción de información en muchos idiomas. *GATE* consta de tres tipos de recursos:

A. *Language Resource* (LR): representa entidades lingüísticas como: lexicones, corpus, ontologías, tesauros, diccionarios.

B. *Processing Resource* (PR): representa entidades algorítmicas como: analizadores sintácticos (*parsers*), generadores, lematizadores, traductores y reconocedores de discurso.

C. *Visual Resources* (VRs): son componentes usados para la construcción de interfaces gráficas.

3.1.2.2. ANNIE (Sistema de extracción de información)

ANNIE (*A Nearly-New Information Extraction System*) es el sistema de extracción de información distribuido de GATE (Ver **Figura 3)**. Depende de algoritmos de estados finitos y del lenguaje JAPE para hacer anotaciones sobre los textos que procesa. ANNIE consta de componentes que realizan diferentes tipos de anotaciones sobre los textos: segmentador de símbolos (*tokeniser*), diccionario de nombres propios (*gazeteer*), divisor de oraciones (*sentence splitter*), etiquetador gramatical (*part of speech tagger*), etiquetador semántico (*semantic tagger*), co-referencia ortográfica (*ortographic coreference*), co-referencia pronominal (*pronominal coreference*), entre otros[4]. En este artículo explicaremos los componentes de mayor relevancia para la extracción automática de contextos definitorios.

4. Algunos de los nombres que aparecen en castellano son tomados del texto de Zhang Zhixiong *et al*., traducido por Marina Jiménez Piano [34]

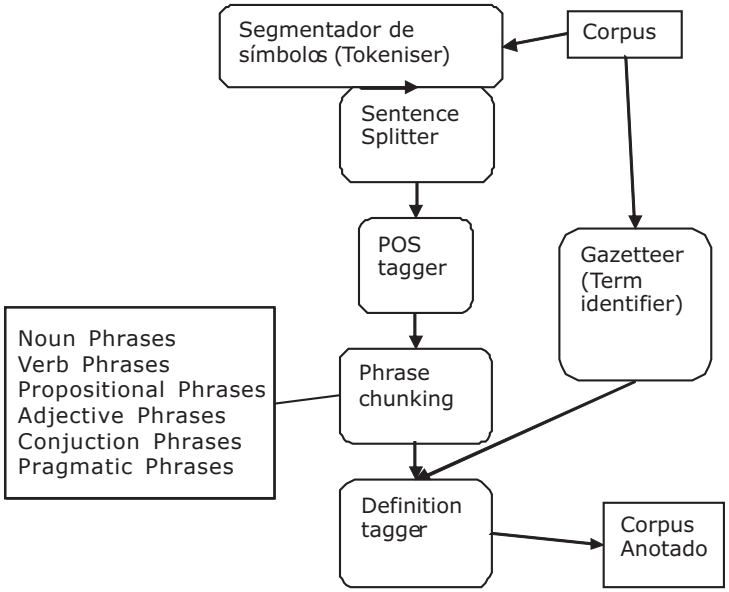

Figura 3. Diagrama de procesamiento con GATE

3.1.2.2.1. *Tokeniser* («segmentador de símbolos»[5])

Esta herramienta realiza una anotación del texto dividiéndolo en símbolos simples (*tokens*), por ejemplo, números, signos de puntuación y distintos tipos de palabras. ANNIE trae, por defecto, unas reglas y tipos de símbolos; éstos pueden especificarse de acuerdo con el uso, según los objetivos de cada proyecto (Ver **Figura 4).**

Figura 4. Resultado del fragmentador de símbolos (*tokeniser*)

5. Proponemos utilizar esta expresión en castellano, para reemplazar el uso del anglicismo *tokeniser*.

3.1.2.2.2. Sentence splitter (divisor de oraciones)

El módulo divisor de oraciones (*Sentence splitter)* consiste en una serie de transductores de estados finitos (patrones JAPE) que dividen el texto en oraciones. Es un módulo requerido para el etiquetador gramatical (*POS tagger*).

3.1.2.2.3. *POS tagger* (etiquetador gramatical)

Este etiquetador produce una anotación de tipo sintáctico sobre cada palabra o símbolo. Con el etiquetador gramatical (*POS-tagger*), finaliza la etapa de procesamiento lingüístico.

3.1.2.2.4. Gazetteer (diccionario de nombres propios)

El diccionario de nombres propios (*Gazetteer*) sirve para etiquetar documentos de acuerdo con las listas definidas en textos planos, con una entrada por línea. Cada lista representa un conjunto de nombres; por ejemplo, de ciudades, de regiones, de países, de entidades (públicas y privadas), etc. En este proyecto, hicimos una correspondencia entre nombres y términos, para lo cual definimos un listado de 1.700 términos en inglés, todos ellos del dominio de las enfermedades neurológicas.

3.1.2.2.5. Phrase chunking (análisis superficial de frases)

Paralelo a la identificación de términos, efectuamos la búsqueda de categorías gramaticales (frases nominales, verbales, preposicionales, etc.); su forma de expresión se realiza en lenguaje JAPE (*Java Annotation Patterns Engine*), que sirve para realizar transducciones de estados finitos sobre anotaciones, de acuerdo con expresiones regulares. Dicho de otro modo, JAPE permite la definición de expresiones regulares, para buscar dentro de las anotaciones y convertirlas en nuevas anotaciones.

3.2. Propuesta de extracción automática de contextos definitorios con GATE

3.2.1. Context Definition Tagger (Etiquetador de contextos definitorios)

Tras la anotación gramatical, se identificaron los patrones verbales y luego los contextos definitorios. Para ello, partimos de una lista cerrada de patrones verbales (Ver **Tabla 2**):

Patrones verbales (verbal patterns)
has already been shown
is considered
constitutes
have been known
is known
referred to
also called
is known
is defined
has been defined
was defined
is called
is the site
are defined
is the term

Tabla 2. Lista cerrada de patrones verbales

La identificación de los contextos definitorios constituye la sumatoria de todos los pasos descritos anteriormente. Además, cabe anotar que para la identificación de dichos contextos se describieron los patrones lingüísticos en JAPE (Java Annotation Patterns Engine)[6]. Ilustramos dicha descripción mediante el siguiente ejemplo:

The CHQ is a generic instrument for measuring health outcomes in children and assessing functional status and well-being. [Contexto definitorio].

Este contexto definitorio (CD) se expresa, según el etiquetaje sintáctico de GATE, de la siguiente manera:

DT+TD[SIGLA]+DV[VBZ]+NP(DT+JJ+NN+PP[IN+VBG+JJ+NN])+PP(IN+NN+CP [CC+VBG+JJ+NN+CC+NN])+ {.}

Este mismo CD, expresado en JAPE, se representa de la siguiente manera (Ver **Figura 5**):

6. JAPE es un motor para anotaciones hecho en Java. Sirve para realizar transducciones de estados finitos sobre anotaciones de acuerdo con expresiones regulares, es decir que permite la definición de expresiones regulares para buscar dentro de las anotaciones y convertirlas en nuevas anotaciones.

```
Rule: Definition
(
{Token.category == DT}
{SpaceToken.kind == space}
{Lookup.majorType == Term}
{SpaceToken.kind == space}
{Lookup.majorType == DefiningVerb}
{SpaceToken.kind == space}
{NounPhrase2}
{SpaceToken.kind == space}
{PrepositionalPhrase}
{Token.string == "."}
):np -->
:np.Definition = { rule = "Definition" }
```

Figura 5. Contexto definitorio expresado en JAPE

Para lograr una mayor expresividad, esto es, que dentro de un patrón se integren muchas definiciones, es importante comparar dichos patrones, de tal manera que puedan extraerse regularidades y representarlas a través de *expresiones regulares*[7].

4. Análisis y resultados

4.1. Construcción del agente deliberativo con la metodología GAIA

Según Wooldridge y Jennings [33], un agente deliberativo o con arquitectura deliberativa es aquel que contiene un modelo simbólico del mundo, explícitamente representado, en donde las decisiones se toman utilizando mecanismos de razonamiento lógico basados en la concordancia de patrones y la manipulación simbólica».

Los agentes deliberativos poseen creencias, deseos e intenciones; estas características se conocen también como «Arquitectura C.D.I. (*Creencias, Deseos e Intenciones*)[8]». La **Figura 6** ilustra estos componentes y sus relaciones.

Ahora bien, GAIA es una metodología utilizada para el desarrollo de aplicaciones bajo el paradigma de agentes; juega un papel en la construcción de estos sistemas,

7. Para la identificación de símbolos que corresponden a *tags* (etiquetas), o marcas hechas por el anotador sintáctico, consultar la siguiente dirección: http://gate.ac.uk/sale/tao/splitap5.html#x24-518000E
8. Expresión equivalente a BDI Architecture (*Belief, Desire and Intention* Architecture).

ya que proporciona un conjunto de pasos incrementales. Se divide principalmente en dos fases: la primera es la de análisis y la segunda la de diseño.

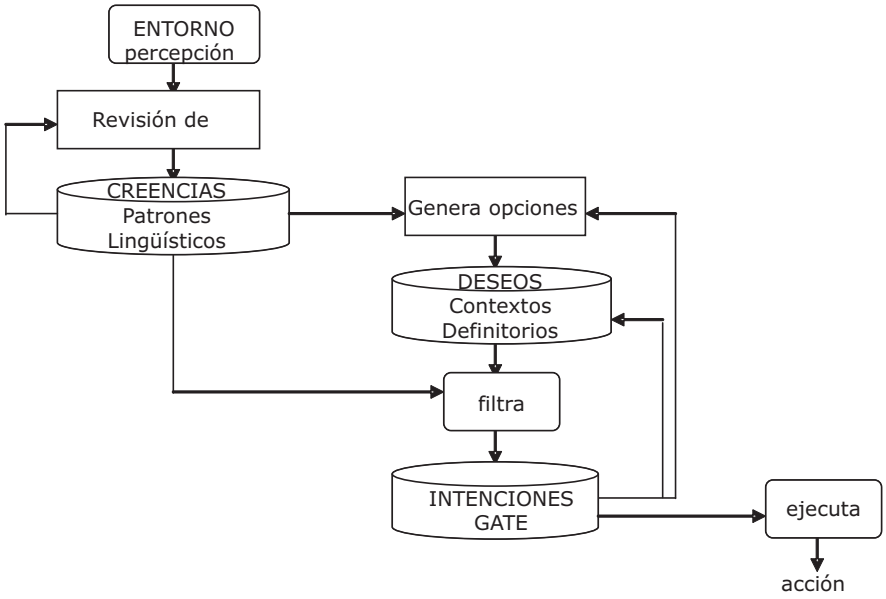

Figura 6. Componentes del Agente Deliberativo

4.1.1. Primera fase de análisis GAIA

El objetivo en esta fase es comprender el sistema y su estructura, sin llegar a referenciar ningún detalle de implementación. (Ver **Tabla 3**)

Rol Usuario
Descripción: quien asume este rol (la máquina), solicita a los usuarios humanos un conjunto de datos para el posterior funcionamiento del sistema.

Funciones·
• Enviar mensajes al agente deliberativo, de acuerdo con las solicitudes del usuario.
• Proporcionar información como URL, lista de los corpus disponibles·
• Mostrar los resultados de los mensajes enviados por el agente deliberativo.

Rol Deliberativo
Descripción: quien asume este rol (de nuevo la máquina) realiza los procesos para encontrar contextos definitorios en el corpus y enviar los resultados al agente usuario.

Funciones
• Identificar patrones lingüísticos en los corpus.
• Consultar a entidades u otros agentes para incrementar los datos de los corpus.
• Verificar si los datos solicitados son válidos y almacenarlos en el corpus.
• Recibir información del agente usuario.
• Buscar nuevos corpus en las direcciones electrónicas (URL) obtenidas por el rol usuario.
• Procesar los mensajes y extraer los parámetros de búsqueda.

- **Modelo de interacciones:** consiste en un conjunto de definiciones de protocolos, uno para cada tipo de interacción entre los roles. Una definición de protocolo consta de los siguientes atributos:
  - Propósito: describe la naturaleza de la interacción.
  - Iniciador: inicia la interacción.
  - Receptor: interactúa con el iniciador.
  - Entradas: usa información durante la interacción (por parte del iniciador).
  - Salidas: suministra información durante la interacción (por parte del receptor).
  - Procesamiento: descripción de todo el cálculo que el iniciador realiza durante la interacción.

Protocolo: «enviar corpus»

Propósito: enviar los corpus al rol deliberativo.	
Rol iniciador: usuario	*Rol receptor:* deliberativo
Entradas: URL, lista de corpus	
Salidas: true	

Protocolo: «extraer contextos definitorios»

Propósito: extraer contextos definitorios.	
Rol iniciador: usuario	Rol receptor: deliberativo
Entradas: URL, lista de Corpus	
Salidas: contextos definitorios	
Procesamiento: el rol deliberativo recibe los textos del corpus, los verifica y los etiqueta; luego, compara con los patrones lingüísticos y extrae los contextos definitorios.	

Tabla 3. Primera fase de análisis GAIA

4.1.2. Segunda fase de análisis GAIA

El objetivo de esta fase es transformar los modelos creados en la fase de análisis, en modelos detallados, y aplicar técnicas clásicas de diseño que lleven a la implementación del sistema.

Con base en lo anterior, el modelo de agentes se representa mediante la siguiente **Figura 7**:

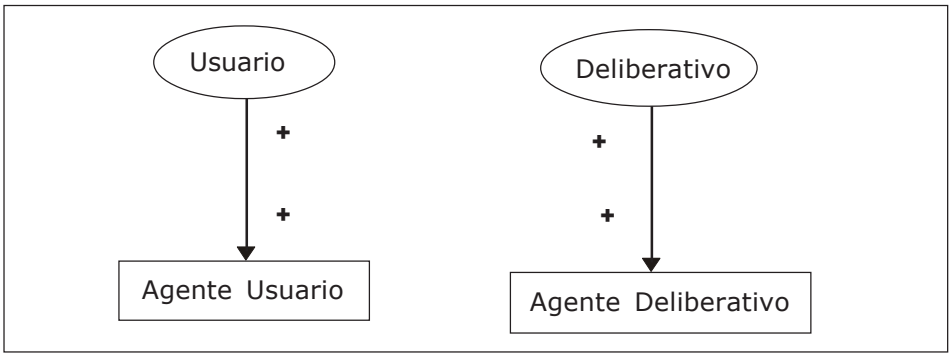

Figura 7. Modelo de agentes del sistema.

- **Modelo de servicios:** en este modelo se identifican los servicios asociados a cada rol de agente y la respectiva especificación de las principales propiedades de dichos servicios. (Ver **Tabla 4**)

Servicio del agente deliberativo: recibir corpus
Entradas: corpus
Salidas: archivo.html
Precondiciones: *true*
Poscondiciones: *true*

Servicio del agente deliberativo: devolver contextos definitorios
Entradas: patrones lingüísticos, corpus
Salidas: archivo.html, contextos definitorios
Precondiciones: *true*

Tabla 4. Modelo de servicio

- **Modelo de familiaridad:** en este modelo se definen los enlaces de comunicación que existen entre los agentes (Ver **Figura 8**).

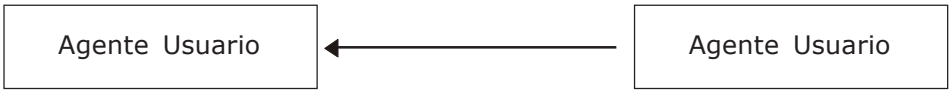

Figura 8. Modelo de familiaridad

Una vez se han analizado y diseñado los agentes, se procede a implementar los agentes en la plataforma de desarrollo JADE.

4.2. Implementación del agente deliberativo

En este apartado se ilustra la implementación del agente deliberativo a través del diseño de las diferentes interfaces. El diseño y la implementación del agente deliberativo mediante las interfaces constituyen el resultado de esta primera fase de la investigación. El agente deliberativo interactúa con el corpus que ha sido analizado previamente con la herramienta GATE, y, como resultado de dicha interacción, extrae los contextos definitorios de manera semiautomática.

*4.2.1. Interfaz básica de aplicación: interacción corpus-agente delibe***rativo**

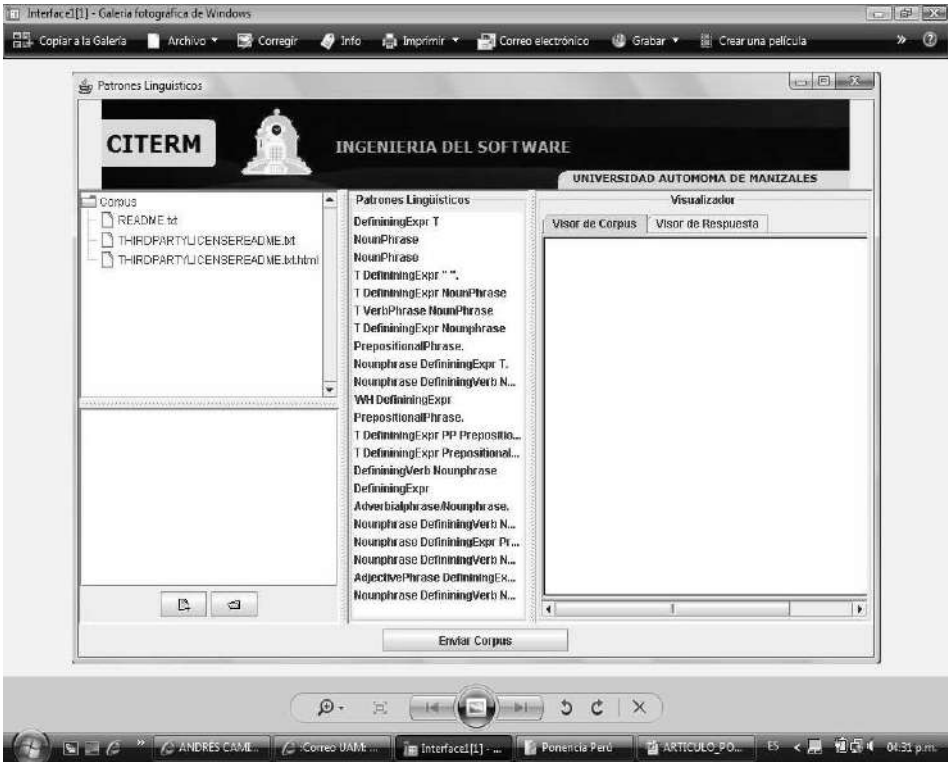

Figura 9. Interfaz básica de la aplicación

La figura 9, se presenta la interfaz básica de la aplicación. En la columna izquierda de la interfaz se observa la lista de los textos que conforman el corpus; éstos se seleccionan y pueden visualizarse en la columna de la derecha, denominada visualizador, el cual contiene dos campos: *visor de corpus* y *visor de respuesta*. En la columna del medio aparecen los patrones lingüísticos, los cuales se aplican a los textos del corpus seleccionado. En la parte inferior se observa el campo **enviar**

corpus, que inicia la interacción con el agente deliberativo. Cuando la aplicación, representada a través de esta interfaz, inicia este proceso, se envían los corpus seleccionados a un agente, se aplican los patrones lingüísticos, con el fin de hallar los contextos definitorios.

4.2.2. Interfaz en interacción con la plataforma JADE

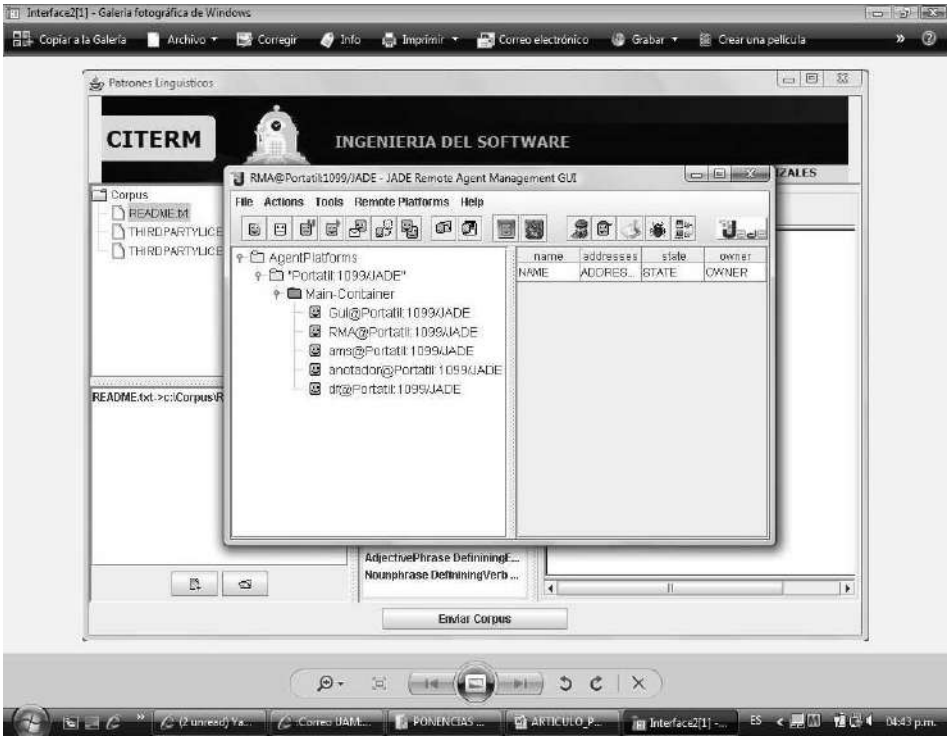

Figura 10. Interfaz en interacción con la plataforma JADE

En la **Figura 10** se observa la interfaz con la plataforma JADE sobrepuesta. En la plataforma podemos identificar los agentes que realizan los procesos de la aplicación «Gui» y «anotador». El agente «Gui» representa la interfaz de la aplicación, y el agente «anotador» es el encargado de hallar los posibles contextos definitorios.

4.2.3. Interfaz de comunicación entre agentes

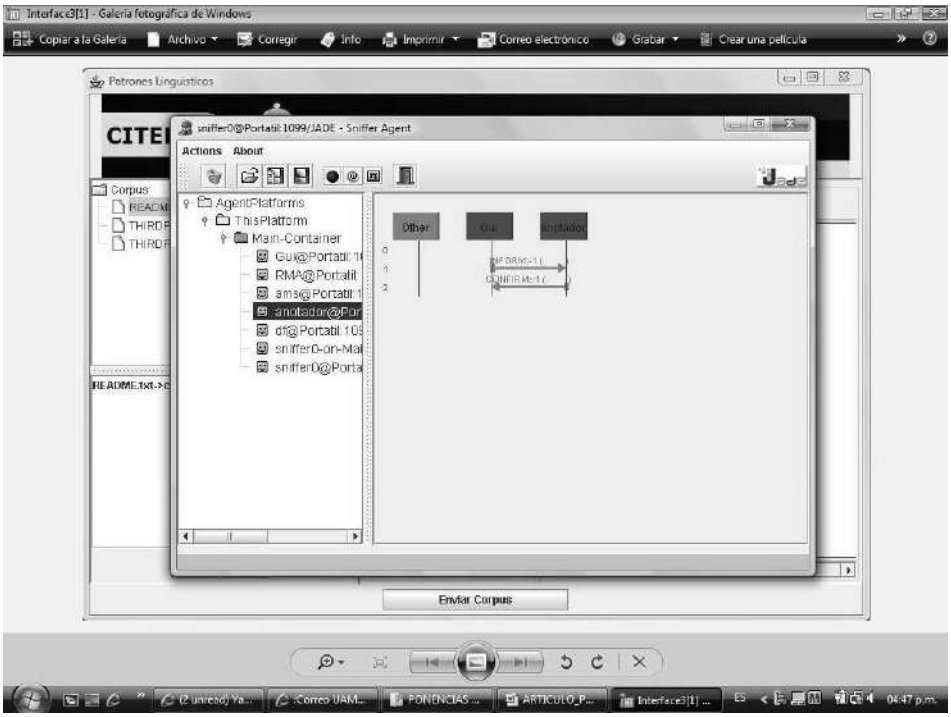

Figura 11. Interfaz de comunicación entre agentes

La **Figura 11** muestra la comunicación y el paso de mensajes ACL entre los 2 agentes; dicha comunicación se da gracias al *sniffer* que tiene la plataforma JADE, para realizar la verificación del proceso.

5. A manera de conclusión - primera fase de la investigación

En este artículo, y según el objetivo planteado para el desarrollo de esta fase del proyecto, se ha logrado presentar el análisis, diseño e implementación de un agente deliberativo para identificar contextos definitorios (CD) en textos especializados.

En lo que concierne a los agentes, se ha podido corroborar que éstos constituyen un nuevo paradigma, que se acopla fácilmente a las nuevas necesidades de la programación distribuida. En cuanto a los Sistemas Multiagente, se ha constatado que permiten modelar los problemas computacionales como entidades que interaccionan para buscar objetivos comunes; y, en relación con los agentes

deliberativos, consideramos que constituyen un avance significativo en el campo de la automatización de la extracción y recuperación de información en textos especializados.

De otra parte, desde la perspectiva de la extracción de los contextos definitorios y de los patrones lingüísticos, se observa que los trabajos realizados hasta el momento no brindan información detallada y desglosada que permita a la máquina reconocer dichos patrones a determinados niveles y refinar la búsqueda con un número de fases más restringido. De todos modos, según los resultados obtenidos, se observa que es posible lograr una representación formal por medio de la descripción de patrones lingüísticos que contribuyan a la extracción automática de CD.

Desde el punto de vista metodológico se ha constatado que la metodología GAIA permite extraer elementos principales de análisis y diseño, los cuales facilitan la definición del agente usuario y del deliberativo.

Por último, a manera de síntesis y desde un punto de vista más descriptivo, puede afirmarse que en esta primera fase del proyecto se realizó la descripción de una serie de patrones lingüísticos a partir de un corpus del dominio de las enfermedades neurológicas, lo cual permitió llegar a una representación formal para la extracción de CD. Igualmente se logró implementar el agente buscador sobre la plataforma JADE, de tal manera que arrojara como resultado archivos que pueden servir como corpus iniciales, para generar la extracción de definiciones del sistema global.

6. Agradecimientos

Este trabajo forma parte del proyecto titulado *Sistema Multiagente para la extracción automática de contextos definitorios basado en ontologías para la Web semántica*, que se realiza gracias a la subvención de las instituciones COLCIENCIAS – UAM con código 1219-405-20249.

Referencias bibliográficas

1. WOOLDRIDGE et al. The GAIA methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 2000, vol. 3, no. 3, p. 285-312.

2. HUANG, Wei; EL-DARZI, E.; JIN, Li. Extending the GAIA Methodology for the Design and Development of Agent-based Software Systems. En: *Computer Software and Applications Conference*, 2007, vol. 2, no. 24-27, p. 159-168.

3. RAJA, A.; LESSER, V. Meta-level reasoning in deliberative agents : intelligent agent technology. En: *Proceedings. IEEE/WIC/ACM International Conference,* Sep.-Dec. 2004.

4. IGLESIAS, C.A. *Definición de una metodología para el desarrollo de sistemas multiagentes,* (Tesis Doctoral). Departamento de Ingeniería de Sistemas Telemáticos de la Universidad Politécnica de Madrid, Madrid: 1998.

5. PA PA, Nyunt; NI LAR, Thein. Software agent oriented information integration system in semantic web. En: *Information and Telecommunication Technologies, 2005. APSITT 2005 Proceedings. 6th Asia-Pacific Symposium.* p. 226-271.

6. SHAFIG, M.O.; DING, Ying; FENSEL, D. Bridging multi agent systems and web services: towards interoperability between software agents and semantic web services. En: *Enterprise Distributed Object Computing Conference*. Oct. 2006, p. 85–96.

7. CHANGJIAN, Deng; YAO, Lan. Architecture of knowledge retrieval based on multi-agent systems. En: *Knowledge Acquisition and Modeling Workshop International Symposium*. 2008, no. 21-22, p.1083-1086.

8. SHAW, M.J.; HARROW, B.; HERMAN, S. Distributed artificial intelligence for multi-agent problem solving and group learning. En: *System Sciences, 1991. Proceedings of the Twenty-Fourth Annual Hawaii International Conference.* 1991, p.13- 26.

9. HUHNS, M. N. y SINGH, M. P. D*istributed artificial intelligence for information systems, CKBS-94 Tutorial*. UK: University of Keele, 1994.

10. MITKAS, P.A.; SYMEONIDIS, A.L.; ATHANASIADIS, I.N. A retraining methodology for enhancing agent intelligence. En: *Integration of Knowledge Intensive Multi-Agent Systems, International Conference,* 2005. p. 422 – 428.

11. SHOHAM, Y. Agent-oriented programming. *Artificial Intelligence*, 1993, vol. 60, no. 1, p. 51–92.

12. BATES et al. *An architecture for action, emotion, and social behavior*. Pittsburgh, PA : Carnegie-Mellon University, 1992. (Technical Report, School of Computer Science ; CMU–CS–92–144).

13. BATES, J. The role of emotion in believable agents. En: *Communications of the ACM*, 1994, vol. 37, no.7, p. 122-125.

14. MAES, P. Agents that reduce work and information overload. *Communications of the ACM*, 1994, vol. 37, no. 7, p. 31–40.

15. ACAY, D.L.; TIDHAR, G.; SONENBERG, L. Extending agent capabilities : tools vs. agents; En: *Web Intelligence and Intelligent Agent Technology International Conference*, 9-12 2008. p. 259–265.

16. FIPA. *Foundation for intelligent physical agents*. [En línea]. Disponible en: http://www.fipa.org. [Consulta: enero de 2009]

17. TECUCI, G. Building intelligent agents: an apprentice ship multistrategy learning theory, methodology, tool and case studies. *Academic Press*, 1998.

18. CHANGJIAN, Deng; YAO, Lan. Architecture of knowledge retrieval based on multi-agent systems. En: *Knowledge Acquisition and Modeling Workshop International Symposium*. 21-22 December 2008, p. 1083-1086.

19. GANDON, F.L. Combining reactive and deliberative agents for complete ecosystems in infospheres. En: *Intelligent Agent Technology, International Conference*._13-16 de octubre de 2003. p. 297 – 303.

20. CORCHADO, J. M. y MOLINA, J. M. *Introducción a la teoría de agentes y sistemas multiagente*. España: Edite Publicaciones Científicas, 2002.

22. RAO, et al. BDI agents from theory to practice. En: *Proceedings of the first International Conference on Multiagent Systems*. 1995. San Francisco USA. p. 312-319.

23. ALARCÓN, R. y SIERRA, G. Reglas léxico-metalingüísticas para la extracción automática de contextos definitorios. En: *Avances en la Ciencia de la Computación, VII Encuentro Nacional de Ciencias de la Computación.* San Luís Potosí: MSCC, Hernández, A., Zechinelli, J.L. (eds). p. 242-247.

24. PEARSON, J. *Terms in context*. Amsterdam: John Benjamins Publishing Co. 1998.

25. MEYER, I. Extracting knowledge-rich contexts for terminography: a conceptual and methodological framework. En: BOURIGAULT, D., Christian Jacquemin and Marie-Claude L'Homme (eds.), *Recent advances in computational terminology*, 2001, vol. 18, 380 pp. (p. 279–302).

26. KLAVANS, J.; MURESAN S. Evaluation of DEFINDER: a system to mine definitions from consumer- oriented medical text. En: *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*. New York: ACM Press. 2000.

27. SAGGION, H. Identifying definitions in texts collections for question answering. *En: International Conference of Language Resources and Evaluation, Proceedings.* Sheffield: England University of Sheffield, Department of Computer Science. 2004.

28. RODRÍGUEZ, C. *Metalinguistic information extraction from specialized texts to enrich computational lexicons*. Barcelona: Universitat Pompeu Fabra. 2004.

29. MALAISÉ et al. Mining defining contexts to help structuring differential ontologies. *Terminology,* 2005, vol. 11, no. 1, p. 21-53.

30. MARSHMAN et al. French patterns for expressing concept relations. *Terminology,* 2002, vol. 8, no. 2, p. 1–29.

31. CONDAMINES, A. Corpus analysis and conceptual relation patterns. *Terminology,* 2002, vol. 1, p. 141-162.

32. ALARCÓN R. y SIERRA G. El rol de las predicaciones verbales en la extracción automática de conceptos. *Estudios de Lingüística Aplicada,* 2003, vol. 38, p.129-144.

33. WOOLDRIDGE, M.J.; JENNINGS, N. R. Intelligent agents: theory and practice. *Knowledge Engineering Review,* 1995. vol. 10, no. 2, pp. 115-152.

34. ZHANG Zhixiong et al. *Hacia la construcción de un sistema de extracción de información chino como soporte de innovación en los servicios bibliotecarios.* [En línea]. Disponible en: http://archive.ifla.org/IV/ifla72/papers/097-Zhixiong_Sa_Zhengxin_Ying_trans-es.pdf [Consulta: enero de 2009]