

SELECCIÓN Y VALIDACIÓN NUMÉRICA DE UNA ARQUITECTURA
MULTIAGENTE PARA MONITORIZACIÓN DE SALUD ESTRUCTURAL

ANDRÉS FELIPE QUINTERO PARRA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA
Y TELECOMUNICACIONES
BUCARAMANGA
2011

SELECCIÓN Y VALIDACIÓN NUMÉRICA DE UNA ARQUITECTURA
MULTIAGENTE PARA MONITORIZACIÓN DE SALUD ESTRUCTURAL

ANDRÉS FELIPE QUINTERO PARRA

Trabajo para optar al título de
Magister en Ingeniería Área Electrónica

Director:

RODOLFO VILLAMIZAR MEJÍA
Dr. Tecnologías de la Información

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA

2011

*A mis padres por todo el apoyo brindado en los momentos difíciles
A mis amigos porque sin su ánimo nada de esto sería posible*

AGRADECIMIENTOS

El autor expresa sus agradecimientos a:

El Dr. Rodolfo Villamizar, director del proyecto, por su guía para la consecución del mismo

El M Sc. Jhonatan Camacho, gran colaborador y gran amigo, que brindó sus conocimientos para el mejoramiento del programa desarrollado.

El M Sc. Juan Carlos García Ojeda por su gran colaboración en el desarrollo de los algoritmos necesarios.

Los doctores Hugo Andrade y Gustavo Chio por los aportes del proyecto.

TABLA DE CONTENIDO

INTRODUCCIÓN	15
1. MONITORIZACIÓN DE SALUD ESTRUCTURAL	17
1.1 Definición del problema	17
1.2 Objetivos	18
1.2.1 Objetivo general	18
1.2.2 Objetivos específicos.....	18
1.3 Algoritmo de monitorización usado en este proyecto.....	19
1.3.1 Desarrollo de un modelo de referencia.....	20
1.3.2 Generación de casos usando registros de aceleración	20
1.3.3 Generación de casos usando funciones de correlación	21
1.3.4 Extracción de características.....	21
1.3.5 Detección, Localización y Cuantificación de daños.....	21
2. AGENTES INTELIGENTES	22
2.1 Definición	22
2.2 Usos típicos	23
3. METODOLOGÍA DE RAZONAMIENTO BASADO EN CASOS (CBR) PARA MONITORIZACIÓN DE SALUD ESTRUCTURAL.....	27
3.1 Adaptación de la metodología	27
3.2 Índices de error	31
3.3 Validación cruzada en CBR	32
4. ADAPTACIÓN DE LA ARQUITECTURA	33
4.1 Selección de la arquitectura.....	33
4.2 Adaptación de la arquitectura	35
5. VALIDACIÓN NUMÉRICA.....	41
5.1 Caso numérico: Estructura Benchmark UBC.....	41
5.2 Generación de casos.....	43
5.3 Identificación de daños.....	44
5.4 Identificación con un sensor defectuoso.....	47
5.5 Aplicación de la metodología propuesta para un caso numérico específico 48	
6. CONCLUSIONES, RECOMENDACIONES Y APORTES.....	56

6.1	Conclusiones	56
6.2	Aportes	58
6.3	Recomendaciones	58
	BIBLIOGRAFÍA.....	60
	ANEXOS	64

LISTA DE ANEXOS

ANEXO A. TEORÍA DE AGENTES INTELIGENTES.....	64
A.1 Tipos de agente	65
A.1.1 Agente de reflejo simple	65
A.1.2 Agente de reflejo basado en un modelo	66
A.1.3 Agente enfocado a una meta	67
A.1.4 Agente basado en la utilidad:	67
A.1.5 Agente de aprendizaje continuo	68
A.2 Propiedades.....	69
ANEXO B DESCRIPCIÓN DE JADE (JAVA AGENT DEVELOPMENT FRAMEWORK).....	71
B.1 Descripción general de la plataforma Middleware	71
B.2 Mensajes en la plataforma JADE	73

LISTA DE FIGURAS

Figura 1. Monitorización de salud estructural de nivel 3 aplicada a una estructura de n pisos.....	19
Figura 2. Monitorización de salud estructural nivel 3 aplicada a una estructura de n pisos, forma global.....	20
Figura 3 Arquitectura multi-agente para monitorización de salud estructural planteada por Yuan et al.....	26
Figura 4 Vector de características de una estructura con cuatro puntos de sensado	28
Figura 5. Metodología CBR aplicada a un sensor.....	28
Figura 6 Metodología CBR aplicada para obtener el diagnóstico global.....	28
Figura 7 Metodología aplicada para obtener el diagnóstico final	31
Figura 8 Adaptación de la arquitectura de Yuan et al.	36
Figura 9 Densidad espectral de potencia de una aceleración en dirección Y	44
Figura 10 Error de dimensión.....	46
Figura 11 Error de piso	46
Figura 12 Paso 1 de la aplicación de la metodología a un caso numérico.....	49
Figura 13 Paso 2 de la aplicación de la metodología a un caso numérico.....	49
Figura 14 Paso 3 de la aplicación de la metodología a un caso numérico.....	50
Figura 15 Paso 4 de la aplicación de la metodología a un caso numérico.....	51
Figura 16 Operaciones realizadas en un agente de monitorización de estructura (STA)	52
Figura 17 Paso 5 de la aplicación de la metodología a un caso numérico.....	54
Figura 18 Porcentajes de daño de cada elemento identificado.....	55
Figura 19 Modelo BDI de los agentes inteligentes.....	64
Figura 20 Esquema de comportamiento de un agente tipo reflejo simple	66

Figura 21 Esquema de comportamiento de un agente de reflejo basado en un modelo	66
Figura 22 Esquema de operación de un agente enfocado a una meta.....	67
Figura 23 Esquema de operación de un agente basado en la utilidad	68
Figura 24 Esquema de operación de un agente de tipo aprendizaje continuo.....	69
Figura 25 Relación de JADE con otras plataformas basadas en el protocolo FIPA	72
Figura 26 Forma de registro de los agentes con una plataforma específica.....	73

LISTA DE TABLAS

Tabla 1. Valores de ejemplo para elaborar un diagnóstico compuesto.....	30
Tabla 2. Factores de ocurrencia del caso ejemplo.....	30
Tabla 3. Elementos diagnosticados con daño y su probabilidad de ocurrencia	31
Tabla 4. Comparación de las diferentes arquitecturas.....	34
Tabla 5. Frecuencias naturales de la estructura metálica.	41
Tabla 6. Indicadores de falsos positivos y negativos con los grupos e validación cruzada	45
Tabla 7 Promedio de errores con daños en los sensores	48
Tabla 8 Diagnósticos realizados por los STAs en un caso numérico.....	53
Tabla 9 Resultados obtenidos del fusionador de datos para un caso numérico. ...	54
Tabla 10 Diagnóstico final del caso numérico.....	55

RESUMEN

TÍTULO:

SELECCIÓN Y VALIDACIÓN NUMÉRICA DE UNA ARQUITECTURA MULTIAGENTE PARA MONITORIZACIÓN DE SALUD ESTRUCTURAL

AUTOR:

ANDRÉS FELIPE QUINTERO PARRA **

PALABRAS CLAVE:

Agentes Inteligentes, Inteligencia artificial, Monitorización de salud estructural (SHM).

DESCRIPCIÓN:

Se presenta una solución para monitorización de salud estructural adaptando una arquitectura multiagente existente. Se muestra el diagrama general de la arquitectura y se explican sus componentes individualmente. El problema de monitorización de salud estructural es mostrado junto con algunas arquitecturas multiagente para presentar la conexión entre los conceptos. La adaptación se realiza utilizando usando la información de otros problemas ya resueltos por los agentes inteligentes, junto con las especificaciones de un sistema de monitorización de salud estructural. Se propone y explica en detalle una metodología para validar numéricamente el sistema. Cada sección de la arquitectura se explica de forma lógica y computacional para mostrar las diferencias entre los algoritmos locales y los de conexión. Se presenta el modelo numérico Benchmark utilizado para la validación numérica del sistema, el cual fue propuesto por la Asociación americana de ingeniería civil (ASCE). El objetivo principal de la metodología es descentralizar los sistemas de monitorización de salud estructural modernos, garantizando integridad de la información para obtener diagnósticos apropiados y los agentes han demostrado ser una excelente opción para descentralizar sistemas, tomando en cuenta los valores de tiempo de procesamiento, toma de decisiones y resultados obtenidos comparados con los sistemas centralizados. El sistema expuesto en este artículo presenta niveles de detección de daño de 98% de aciertos, siendo estos similares a los obtenidos por los sistemas centralizados. Además presenta confiabilidades del 70% en promedio frente a fallas en un sensor.

* Proyecto de Investigación

** Facultad de Ingenierías Físico Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Rodolfo Villamizar Mejía.

SUMMARY

TITLE:

SELECTION AND NUMERICAL VALIDATION OF A MULTIAGENT ARCHITECTURE FOR STRUCTURAL HEALTH MONITORING

AUTHOR:

ANDRÉS FELIPE QUINTERO PARRA **

KEYWORDS:

Artificial Intelligence (AI), Intelligent agent, Structural Health Monitoring (SHM).

DESCRIPTION:

This paper presents an approach for structural health monitoring (SHM) by the adaptation of an existing multi-agent architecture. The architecture's general scheme is shown and its components are explained individually. The structural health problem is presented briefly along with some multi-agent architectures in order to present the connection between concepts. The adaptation is made using the information from previous problems solved by intelligent agents and the structural health monitoring system specifications. A methodology is proposed and explained for a numerical validation of the system. Each section of the architecture is explained in a computational way to make a difference between algorithms and agents' connections. The numerical Benchmark model used for the system's numerical validation is presented. This model was proposed by the ASCE and is used to test SHM systems. The methodology is proposed in order to decentralize the current SHM systems as much as possible, in order to guarantee data integrity. Also, intelligent agents have been used in other monitoring problems and they have demonstrated to be an excellent solution due to processing time, decision making and general behavior in comparison to centralized systems. The system presents high levels of damage detection when cross-validated reaching 98% of accurate detection, thus obtaining similar behavior to centralized structural health monitoring systems. Also the system presents 70% of reliability when a sensor fails, surpassing current SHM systems.

* Research project

** Physical and Mechanical Engineering Faculty, Electrics, Electronics and telecommunications School. Rodolfo Villamizar Mejía.

INTRODUCCIÓN

Los daños en estructuras civiles han causado innumerables desastres a lo largo de la historia, afectando la salud de las personas y su economía. Por lo anterior, la comunidad científica del área de estructuras se ha interesado en el estudio de metodologías y técnicas que busquen predecir y/o evitar tales daños.

En todo momento los edificios y puentes se están deteriorando debido al paso de tiempo, perdiendo así su utilidad y beneficio para el ser humano, y de esta premisa surge la pregunta ¿Existe alguna forma de conocer estos daños para prevenirlos, corregirlos y conocer el tiempo de vida útil que posee cada estructura? Para responder a este cuestionamiento se ha desarrollado un área de investigación conocida como Monitorización de Salud Estructural (Structural Health Monitoring o SHM) que busca evaluar las condiciones dinámicas de las estructuras para determinar, localizar, cuantificar daños en la misma y además predecir la vida útil de la misma. [1]

Para obtener las características que indican el estado de salud de grandes estructuras, se requiere de un sistema de monitorización, ya que cuantificar propiedades físicas de manera visual o con métodos acústicos es complejo. Desde el punto de vista de la ingeniería estructural, los sistemas de monitorización deben ser operados remotamente y deben poseer un procesamiento de datos in situ, así como una forma sencilla y simple de instalación y mantenimiento. También deben poseer las técnicas de detección de daños incorporadas y un sistema de alerta en caso de una falla de energía junto con una base de datos. Los sistemas de monitorización actuales están basados principalmente en sistemas expertos que reúnen toda la información disponible de la estructura en un punto central. [2][3]

Un defecto de los sistemas centralizados es que no presentan ningún resultado cuando no cuentan con la información completa de la estructura, ya que no pueden inferir resultados ni distribuir su función en subsistemas. Para solucionar este problema se plantea utilizar una arquitectura multi-agente, compuesta por

diversos agentes inteligentes capaces de operar autónomamente y tomar decisiones independientes o conjuntas [4], adaptada para aplicar razonamiento basado en casos con el fin de detectar, localizar y cuantificar cambios de rigidez en una estructura civil.

El [capítulo 1](#) presenta una revisión de las técnicas utilizadas en monitorización de salud estructural y el algoritmo utilizado en este proyecto. El [capítulo 2](#) presenta los agentes inteligentes con su definición y sus aplicaciones. El [capítulo 3](#) muestra el razonamiento basado en casos aplicado a monitorización de salud estructural. El [capítulo 4](#) presenta la selección y adaptación de una arquitectura multi-agente para dar una solución al problema SHM. El [capítulo 5](#) presenta el caso numérico sobre el cual se validó la arquitectura, así como las pruebas realizadas y los indicadores de error. En el [capítulo 6](#) se presentan las conclusiones del trabajo con los agentes, así como recomendaciones para trabajos futuros. Posteriormente se incluye la bibliografía respectiva. En el [anexo A](#) se presenta la información referente a los agentes inteligentes y sus tipos. El [anexo B](#) presenta el protocolo ACL y una explicación de JADE.

MONITORIZACIÓN DE SALUD ESTRUCTURAL

1.1 Definición del problema

La monitorización de salud estructural se ha trabajado desde finales del siglo XIX, donde los trabajadores de las vías escuchaban las emisiones acústicas de la misma a fin de detectar fallas o grietas. Estas técnicas de reconocimiento visual o auditivo son las alternativas más utilizadas cuando no se cuenta con tecnologías avanzadas de medición y cómputo que permiten realizar una monitorización más adecuada. Dado al avance computacional de los últimos treinta años, se han desarrollado varias técnicas basadas en principios físicos y algoritmos de procesamiento de la información con mayor capacidad que los realizados en épocas anteriores [1] Existen cinco pasos en un proceso de monitorización de salud estructural:

- 1) Inspección visual
- 2) Modelado matemático de la estructura
- 3) Instrumentación de la estructura
- 4) Desarrollo de técnicas de evaluación de daños
- 5) Validación e implementación de las técnicas para la monitorización de daño

Usualmente el proceso de la inspección es el más utilizado para monitorizar, ya que los daños revelados en grietas son bastante evidentes y sirven para poder determinar la efectividad de otras técnicas. Sin embargo no revela detalles acerca de los daños internos de la estructura y está sujeto a errores por parte del observador. [5]

La mayoría de las implementaciones actuales de Monitorización de salud estructural están basadas en sistemas expertos [1]. Sin embargo, de acuerdo con [6], un sistema centralizado es más lento y presenta menos adaptabilidad que los

descentralizados. Una solución apropiada para abordar este problema es utilizar una técnica basada en sistemas distribuidos, ya que la información relevante de la estructura puede estar contenida en puntos distribuidos geográficamente. Entonces, dado que los sistemas multi-agente son sistemas distribuidos con una arquitectura ajustada para cada problema específico, se puede utilizar una arquitectura adaptada para el problema de monitorización. Dado lo anterior, se necesita un estudio de las arquitecturas existentes para seleccionar y adaptar la más adecuada para monitorizar la salud de estructuras civiles.

De acuerdo a la información disponible, la hipótesis de esta propuesta fue: “Una de las arquitecturas actuales puede ser ajustada para utilizarla en monitorización de salud estructural y puede comportarse de una mejor forma que las soluciones actuales en monitorización, por lo menos para un modelo estructural específico”. Para probarla se estudiaron las soluciones actuales del problema de monitorización estructural y las arquitecturas multi-agente, con lo que se seleccionó una para dar realizar monitorización descentralizada y se validó con un caso numérico.

1.2 Objetivos

1.2.1 Objetivo general

Seleccionar y adaptar una arquitectura multi-agente para monitorización de salud estructural, evaluada numéricamente para un modelo Benchmark de ingeniería civil.

1.2.2 Objetivos específicos

1. Seleccionar una arquitectura multi-agente apropiada para monitorización de salud estructural, teniendo en cuenta los requerimientos del problema.
2. Adaptar la arquitectura seleccionada tomando en cuenta sus parámetros de configuración y los requerimientos de la monitorización de salud estructural.
3. Validar numéricamente la arquitectura adaptada usando un modelo Benchmark de una estructura civil.

1.3 Algoritmo de monitorización usado en este proyecto

La metodología planteada en el presente trabajo se basa en los 4 módulos del reconocimiento de patrones de acuerdo con [5] adaptada de acuerdo a los requerimientos del problema de monitorización estructural descentralizado.

Los 4 módulos del Reconocimiento de Patrones pueden ser reemplazados por 3 etapas establecidas por las condiciones dadas para un problema Benchmark y por los algoritmos disponibles. De esta forma se obtienen tres módulos base los que se replican a nivel local y posteriormente a nivel global, para obtener un diagnóstico más acertado.

Dado que la efectividad de muchas técnicas está basada en el pre-procesamiento de las señales, se han desarrollado varias técnicas para detectar daños adecuadamente utilizando la mínima cantidad de información disponible para economizar recursos computacionales. [1] [2]

Como la descentralización de la información requiere una fusión de la información para obtener un diagnóstico apropiado, se propuso utilizar algoritmos de recopilación de datos. La **¡Error! La autoreferencia al marcador no es válida.** muestra el algoritmo para la obtención de diagnósticos independientes acerca de la salud estructural de un edificio.

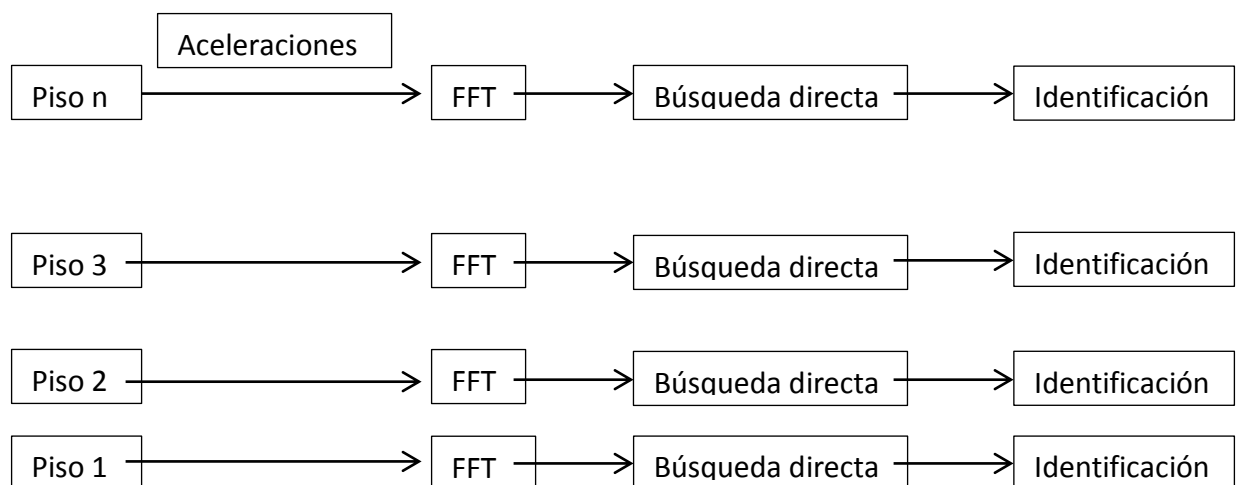


Figura 1. Monitorización de salud estructural de nivel 3 aplicada a una estructura de n pisos.

Cada piso identifica cuales son las variaciones porcentuales de rigidez de la estructura, basadas en la información con la que cuenta. Sin embargo, los diagnósticos locales van a un punto central donde se realiza la identificación de daños total como se puede apreciar en la Figura 2

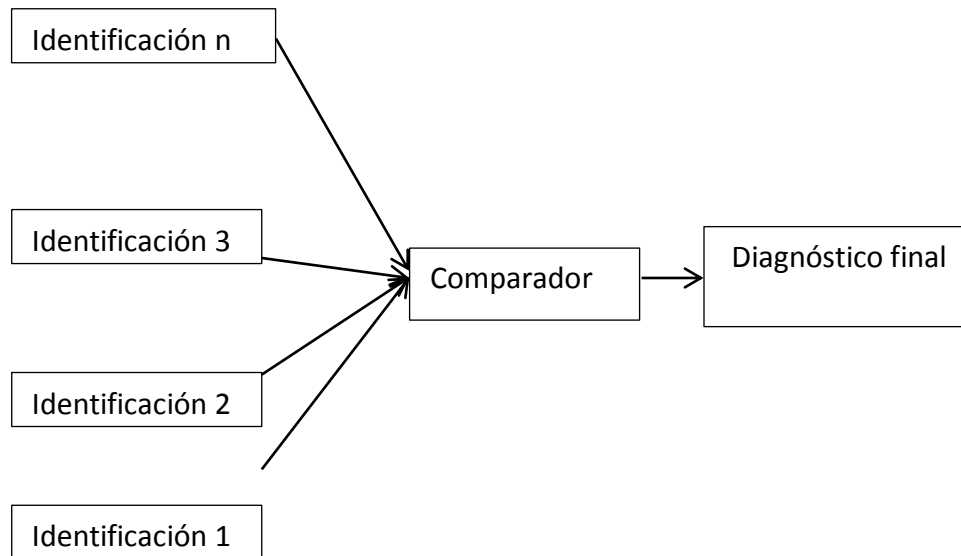


Figura 2. Monitorización de salud estructural nivel 3 aplicada a una estructura de n pisos, forma global.

1.3.1 Desarrollo de un modelo de referencia

Se utiliza el modelo de referencia planteado por [5] que representa una estructura en función de las matrices de masa, amortiguamiento y rigidez.

1.3.2 Generación de casos usando registros de aceleración

A continuación se generan una base de casos a partir de variaciones porcentuales de la rigidez nominal de la estructura. Los registros de aceleración se utilizan para generar la base de casos cuando la excitación de la estructura es conocida (vibración forzada) utilizando la transformada de Fourier para obtener sus propiedades en frecuencia.

1.3.3 Generación de casos usando funciones de correlación

Siguiendo las indicaciones planteadas por [5], se utiliza la técnica de excitación natural (NExT) para obtener la respuesta dinámica de la estructura debida a perturbaciones ambientales.

1.3.4 Extracción de características

Como se quieren encontrar los efectos de las variaciones porcentuales de rigidez sobre la respuesta natural de la estructura, se utiliza la transformada rápida de Fourier para obtener las frecuencias de la estructura y su amplitud, con lo cual se construye un vector de características que contiene la amplitud y posición de cada frecuencia natural para compararlos con el vector de características del edificio en estado sin daño.

1.3.5 Detección, Localización y Cuantificación de daños.

Para la etapa de entrenamiento se utilizan bases de datos locales con el fin de corroborar que el sistema encuentra la región afectada por el daño correctamente, y reconoce el caso sin daño.

Para la etapa de validación se utiliza un nuevo caso, es decir se excita al sistema de forma diferente a los casos de entrenamiento y se busca en el sistema si la respuesta combinada fue acertada o el sistema no pudo dar un diagnóstico adecuado.

AGENTES INTELIGENTES

2.1 Definición

Los agentes inteligentes son programas caracterizados por ser autónomos, poseer capacidad de comunicación, pro-actividad e introspección.[4][7][8]. (Ver anexo A). Cuando se reúnen varios agentes para llevar a cabo una función determinada, el grupo se conoce como Arquitectura Multi-agente ya que posee grupos que realizan tareas independientes con una meta global única.[9] Existen diferentes tipos de arquitecturas clasificadas según su relación con el entorno, que son: Reactiva, Lógica e Híbrida. [10]

- **Reactiva:** Se compone de agentes principalmente reactivos que no realizan toma de decisiones ni análisis de variables, sino que responden ante un estímulo externo con acciones predeterminadas sin capacidad de anticipación ni control de daños.
- **Lógica:** Trabaja con datos numéricos exclusivamente y no recibe información del entorno por lo que son útiles para experimentar reacciones de cada agente ante eventos programados.
- **Híbrida:** Recibe estímulos del entorno y realiza procesos de toma de decisiones en sus agentes para elaborar una tarea compleja. Este tipo de arquitectura es uno de los más útiles ya que combina la capacidad de percepción del entorno de los agentes así como su capacidad de introspección y además permite un trabajo descentralizado otorgando un control de daños más eficiente sobre el sistema.

Dado que la interacción entre agentes simula la interacción entre seres humanos, existen diversos tópicos de investigación como lo son las dinámicas de consenso, la solución distribuida de problemas, la planeación a diversas metas, la toma de decisiones a múltiple escala, entre otros. [11]

2.2 Usos típicos

Los agentes inteligentes son usados para controlar el funcionamiento de componentes dentro de una estructura lógica, ya que para esto se requiere un esquema organizado de vigilancia que permita la identificación de alarmas específicas y una correcta interpretación, negociación con el agente dañado para brindar una respuesta al usuario y al agente para corregir la falla. [12]

Las estructuras multi-agente pueden interactuar entre sí de diversas formas de acuerdo a un orden jerárquico o a un orden de funcionalidades específicas. Algunas arquitecturas pueden basarse en roles fijos asignados manualmente tal como la expuesta por [13] o en roles dinámicos que permiten funciones de recolección de datos como la planteada por [14]. Las diferentes arquitecturas pueden cumplir funciones, como la distribución de datos de una forma más eficiente que una red de sensores [8] o el control distribuido de un sistema que permita la ejecución concurrente de todas las acciones si fuese necesario y que asegure una mayor eficiencia del sistema [15].

También son usadas para controlar redes de sensores distribuidas, utilizando las negociaciones entre agentes para determinar los datos correctos e incorrectos [16]. MACRO (Multi-agent Architecture for Coordinated Responsive Observations) es una arquitectura utilizada para afrontar los retos que presenta SEAMONSTER (South East Alaska Monitoring Network for Science, Telecommunications, Education and Research) tales como la ejecución de acciones dependientes del Hardware y los manejos de datos que estas acciones implican. Esta arquitectura es una adaptación de la arquitectura CORBA [17]. JAIN es una arquitectura controlada para simulación de entornos y proyecciones, monitorizando cada variable de entorno. Proporciona un mejor rendimiento que una arquitectura convencional de estructurado.[18] Otro uso típico de las arquitecturas de multi-agentes es para controlar robots que deben realizar tareas conjuntas como en una olimpiada de futbol. Aunque su uso comercial sea casi nulo es conveniente mencionar que utilizan una arquitectura basada en roles en la que cada componente conoce sus capacidades y propone su solución sin ningún nivel

jerárquico establecido, es decir, todas las soluciones son aceptadas y estudiadas con igual número de relevancia por todos los agentes para tomar la decisión más apropiada. [19] [20]

Existen agentes utilizados para simular el comportamiento de los mercados de energía basados en las propiedades de introspección, comunicación y negociación de los agentes y definiendo una arquitectura donde todos ofertan libremente y los moderadores consultan regularmente a los ofertantes para evitar ofertas indebidas.[21] Una arquitectura propuesta en [22] utiliza un agente móvil, el cual es un programa que utiliza al máximo las capacidades de un sistema local y se va desplazando a otros sistemas para utilizar las capacidades de todos y obtener una solución rápidamente. La arquitectura ARCHON es utilizada para el transporte de electricidad y es interesante porque presenta varios agentes de reflejo simple y solo uno con una meta, la cual es detectar el funcionamiento correcto de cada línea de transmisión.[23] La arquitectura JEINA [18] presenta algunas propiedades interesantes. Su función es monitorizar el control de flujo en línea, por lo que siempre está en contacto con el ambiente y procesando la información. Utiliza KQML como protocolo de comunicación. La arquitectura ORG [24] presenta una aproximación interesante para monitorizar las acciones de un vehículo externo, para lo cual se construyó una red distribuida de sensado para poder ejecutar procesos de control organizacional, aprendizaje de los agentes a nivel global y coordinación de cada agente. Sin embargo esta arquitectura está basada en la parte lógica y no ha sido adaptada para el trabajo con la red distribuida de sensores en línea, lo cual dificulta la implementación de algoritmos que requieran transmisión de datos en tiempo real. La arquitectura MAGMA [25] está compuesta de tres niveles para resolver problemas del tipo meta-heurístico. Es interesante observar como plantea el seguimiento de la solución ya que en todo momento está revisando las salidas de los agentes con el fin de descartar aquellas que no estén dentro del rango aceptable.

Las aplicaciones de agentes inteligentes para problemas de monitorización son diversas, sin embargo algunos no presentan el esquema completo de su arquitectura, por lo que replicar el trabajo necesitaría diseñar la arquitectura utilizando metodologías como GAIA [26] o ZEUS [27]. Sin embargo el diseño de una arquitectura completa esta por fuera de los objetivos de este proyecto.

La arquitectura propuesta en [28] utiliza un grupo de sensores para obtener información de la estructura por diversas fuentes, y luego combinarla utilizando herramientas matemáticas. Esta arquitectura ha sido comprobada en una estructura artificial diseñada para tal fin y presenta una buena forma de encontrar los daños, ya que se basan en daños controlados físicamente con una estructura, pero no toman en cuenta la extrapolación a otras estructuras no instrumentadas y no mencionan el pre procesamiento que le realizan a las señales acústicas recibidas. Sin embargo es una arquitectura híbrida ya que recibe datos del entorno y realiza procesos lógicos para calcular los elementos que presentan daño. La estructura utilizada presenta 64 tornillos en donde la falta de uno representa un cambio significativo en la rigidez de la misma. En la Figura 3 se pueden observar todos los tipos de agentes que posee la estructura así como las comunicaciones permitidas entre ellos. [6]

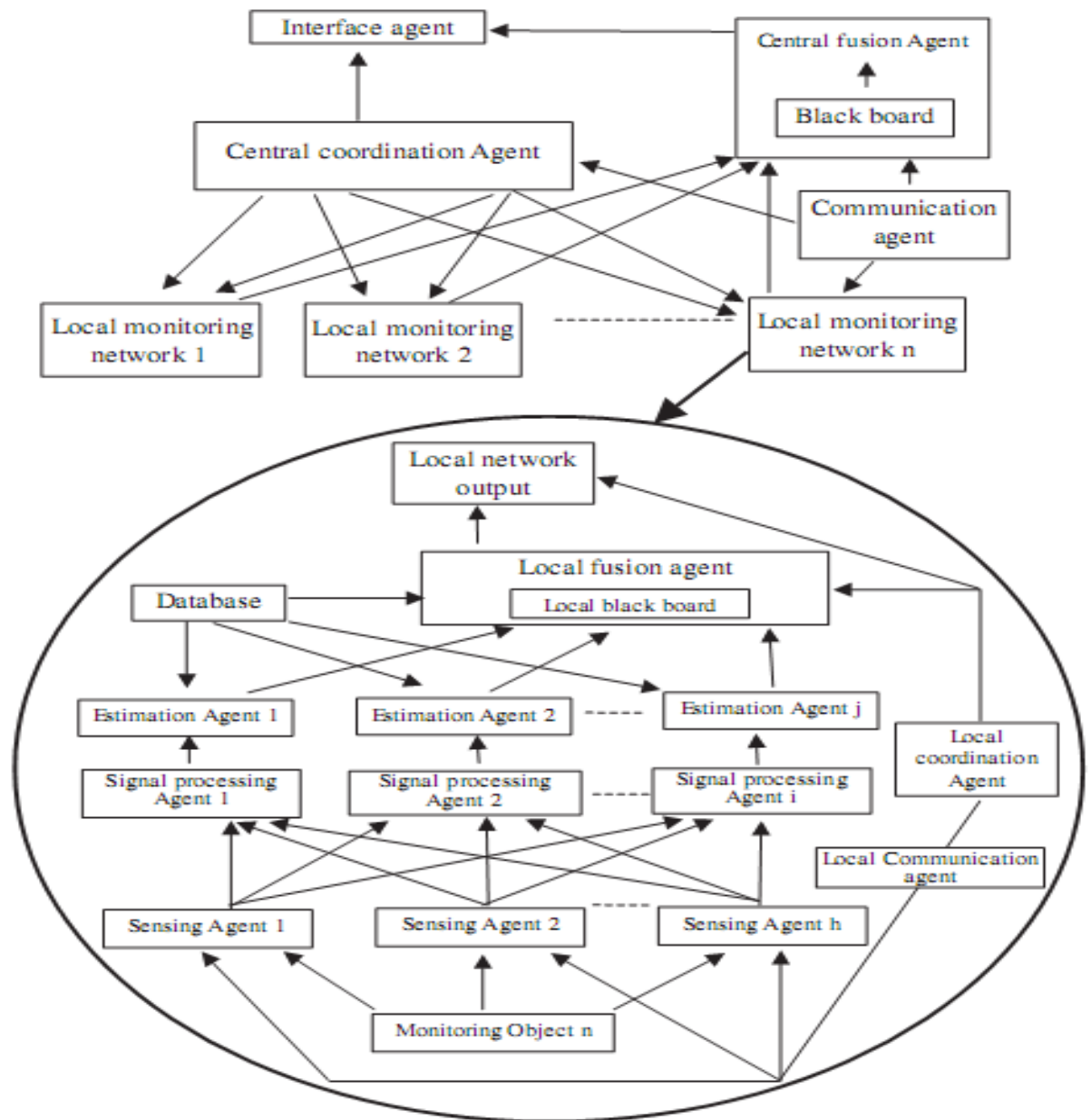


Figura 3 Arquitectura multi-agente para monitorización de salud estructural planteada por Yuan et al.

Fuente [28]

METODOLOGÍA DE RAZONAMIENTO BASADO EN CASOS (CBR) PARA MONITORIZACIÓN DE SALUD ESTRUCTURAL

El problema de algunos sistemas de monitorización de salud estructural es su dependencia de la totalidad de la información para dar su diagnóstico (esquema centralizado). Para abordar este problema, se plantea un esquema descentralizado en donde la monitorización de salud estructural se realiza mediante el razonamiento basado en casos (CBR) independiente por subsistemas locales, dentro de una arquitectura multi-agente.

Para el desarrollo de esta metodología se asume que se cuenta con una base de casos con suficientes registros de aceleración de la estructura, obtenidos a partir de un modelo numérico o medidas experimentales tomadas de la estructura. Esta metodología se basa en la propuesta por Camacho [5] con algunos cambios para poder descentralizar los diagnósticos.

3.1 Adaptación de la metodología

Debido a las propiedades de las señales obtenidas al correlacionar las aceleraciones, se decidió utilizar la transformada rápida de Fourier para obtener la densidad espectral de potencia de la aceleración en dirección Y en cada piso de la estructura. No se utiliza la transformada Wavelet ya que representa un gasto computacional adicional y su información de la señal en el dominio del tiempo no fue utilizado en este proyecto. No se utilizan los métodos de reducción de dimensionalidad ni la red SOM como elemento interprete, ya que, considerando que cada sensor tiene su propio sistema de interpretación de datos, utilizar herramientas computacionales de este tipo requiere de software más especializado que el requerido para efectuar la transformada rápida de Fourier y una búsqueda en una base de casos local. El vector de características utilizado está compuesto por las amplitudes y posiciones de los máximos locales en frecuencias cercanas a las frecuencias naturales de la estructura.

La Figura 4 muestra el contenido de un vector de características y la Figura 5 representa la metodología CBR aplicada a un sensor.

AM= Amplitud del Máximo local PM= Posición del Máximo local

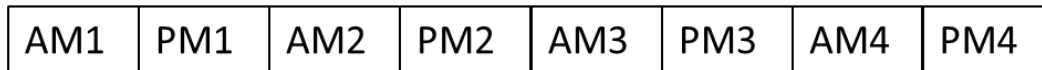


Figura 4 Vector de características de una estructura con cuatro puntos de sensado

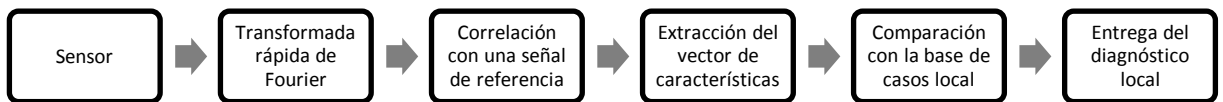


Figura 5. Metodología CBR aplicada a un sensor.

Tras aplicar la metodología a cada sensor, se efectúa un análisis global para poder determinar, con toda la información posible, cual es el diagnóstico más acertado. Para realizarlo se aplica CBR utilizando la información obtenida de todos los sensores como se muestra en la Figura 6. La fusión de la información consiste en concatenar los vectores de características.

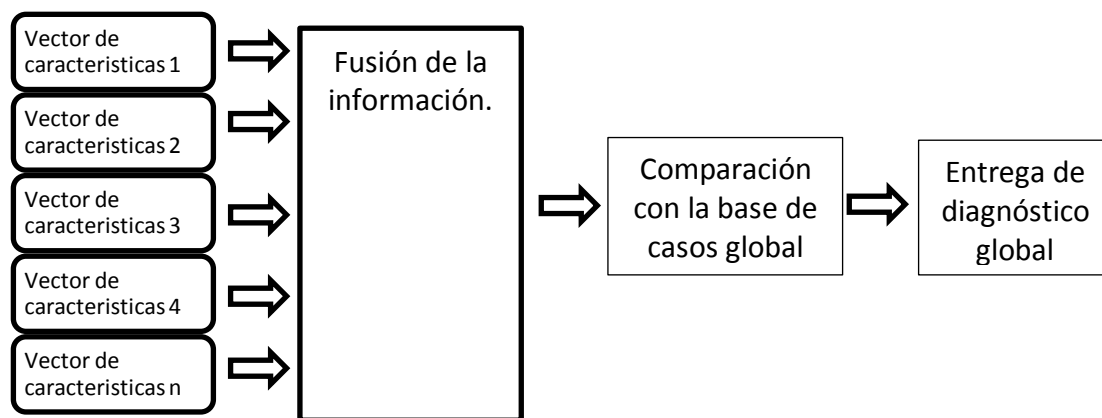


Figura 6 Metodología CBR aplicada para obtener el diagnóstico global

Cada diagnóstico entrega el número del elemento dañado y su probabilidad de ocurrencia, parámetro calculado en función de la distancia euclidiana de cada elemento.

Para obtener el resultado final que interesa al usuario se realizan varios procesos. En primer lugar se determina la dimensión del daño realizando un promedio ponderado entre los diferentes estimados de la misma. El diagnóstico general, que se obtiene con la base de casos global, posee un mayor peso ya que reúne una mayor cantidad de información para realizar su análisis. En general la dimensión se estima de la siguiente manera, donde n es el total de sensores

$$dim_{total} = \frac{dim1 + dim2 + dim3 + \dots + dimn + dim_{general} * 1.2}{n + 1.2}$$

Luego se realiza un promedio de las severidades encontradas por cada sensor, teniendo en cuenta el peso asignado al diagnóstico general. La ecuación de severidad es

$$sev_{total} = \frac{sev1 + sev2 + sev3 + \dots + sevn + sev_{general} * 1.2}{n + 1.2}$$

A continuación se seleccionan los elementos que más se repiten en los diferentes diagnósticos junto con su probabilidad de ocurrencia y se obtiene un nuevo factor llamado factor de ocurrencia el cual está definido de la siguiente manera, donde Poc es la probabilidad de ocurrencia.

$$factor\ elemento\ n = \frac{poc1 + poc2 + \dots + pocn + poc_{general} * 1.2}{n + 1.2}$$

A medida que un elemento aparezca más veces en los diferentes diagnósticos, es más probable que realmente sea un elemento que presente daño significativo para la estructura. Para ilustrar la aplicación de esta parte de la metodología se plantea un caso de ejemplo donde se ha aplicado la metodología a nivel local y se han obtenido los diagnósticos mostrados en la Tabla 1

Tabla 1. Valores de ejemplo para elaborar un diagnóstico compuesto

# Sensor	Elementos dañados	Probabilidad de daño de cada elemento	Severidad
1	12 13	100 - 80	20%
2	14	100	30%
3	13 15	100 - 70	20%
General	13 15	100 - 80	25%

Se calcula la dimensión del daño por medio de la siguiente ecuación

$$dimtotal = \frac{2 + 1 + 2 + 2 * 1.2}{3 + 1.2} = 1,7619$$

Debido a que el valor de dimensión solo puede ser un entero se aproxima al entero más cercano. En este caso la dimensión es 2. Se calcula la severidad del daño en la estructura definido por

$$sevtotal = \frac{20 + 30 + 20 + 25 * 1.2}{3 + 1.2} = 23,8095\%$$

Luego se calculan los factores de ocurrencia, cuyos resultados se muestran en la Tabla 2.

Tabla 2. Factores de ocurrencia del caso ejemplo

# Elemento	Factor
12	23,8095
13	71,4286
14	23,8095
15	39,5238

Como la dimensión es igual a 2, se escogen los elementos que posean mayor factor de ocurrencia. En caso de encontrar más de un elemento con el mismo

factor, se opta por el elemento que haya aparecido en el diagnóstico general. Si todos estos elementos aparecieron en el diagnóstico central y poseen el mismo daño se utiliza el elemento con menor número de identificación.

Tabla 3. Elementos diagnosticados con daño y su probabilidad de ocurrencia

# Elemento	Porcentaje de ocurrencia
13	71,42 %
15	39,52 %

Por lo tanto se obtienen los elementos con más probabilidad de estar dañados en la estructura siguiendo un proceso matemático de exclusión.

En la Figura 7 se muestra la metodología completa para obtener un diagnóstico final que se entrega al usuario. En la figura se asume que ya se obtuvieron los diagnósticos locales y el general aplicando lo descrito anteriormente.

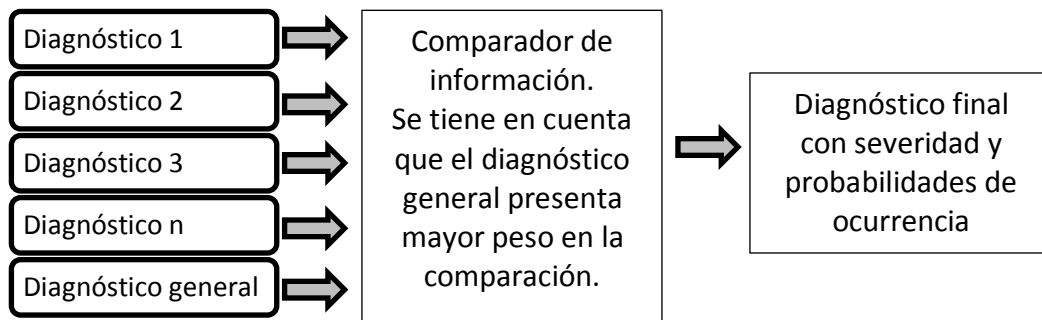


Figura 7 Metodología aplicada para obtener el diagnóstico final

3.2 *Indices de error*

Los indicadores de error utilizados son el error de dimensión, error de severidad, cantidad de falsos positivos y falsos negativos [5]

Se utiliza otro indicador relacionado con la robustez del sistema y está relacionado con la capacidad de detección de daño del sistema cuando uno o más sensores no operan correctamente que se define de la siguiente manera:

$$\text{Detecciones con poca información} = \frac{\# \text{casos detectados correctamente}}{\# \text{casos totales}}$$

Este valor se calcula en el sistema general realizando fallas de uno o más sensores. Los casos detectados correctamente solo incluyen casos en los que el sistema determina cual sección de la estructura está dañada. Debido a la metodología utilizada existen casos donde con un solo sensor se obtiene un diagnóstico apropiado mientras que en otros casos no suceda lo mismo.

3.3 Validación cruzada en CBR

La validación cruzada consiste en dividir aleatoriamente el conjunto de casos de entrenamiento en N grupos. Luego, sobre N-1 grupos se aplica la metodología CBR y se evalúa el error de identificación para el grupo que no fue tenido en cuenta. El proceso se repite de tal forma que a cada grupo se le calcule un error de identificación. De esta forma se obtienen N errores de identificación para los cuales se calcula el valor promedio y la desviación estándar.

Esto se realiza con el fin de poder obtener un indicador de que tan óptimo es el sistema para manejar datos que no estén dentro de su base de casos en el modo operación ya que si el sistema se prueba con los mismos casos que posee su base evidentemente encontrará rápidamente la solución más apropiada. Dos indicadores que pueden ser usados para comparar este sistema con otros que han trabajado monitorización utilizando CBR como [5], es el error promedio y la desviación estándar de la detección de daño a nivel global.

ADAPTACIÓN DE LA ARQUITECTURA

De acuerdo a lo observado en el capítulo anterior, se necesita un sistema que pueda realizar operaciones descentralizadas y que luego pueda fusionar la información. Sin embargo este sistema debe poder determinar cuando falla una de las fuentes de información para evitar que el sistema en general entregue un dato erróneo. Para solucionar este problema se planteó la adaptación de una arquitectura multi-agente que pueda cumplir con esta metodología y se acople a las necesidades de un sistema moderno [2] [29].

Como la monitorización de salud estructural se ha trabajado como un proceso centralizado, normalmente se requiere de un sistema experto para interpretar los datos y coordinar las acciones de los sensores y coordinar la comunicación de datos. En la última década ha aumentado el número de aplicaciones inalámbricas en esta área pero se ha mantenido el uso de un sistema experto coordinador. Sin embargo, se han planteado varias alternativas a este tipo de coordinación, debido a que un sistema centralizado posee menos velocidad, confiabilidad y diversidad de funciones que un sistema distribuido. [6] [28]

Las arquitecturas multi-agentes han sido usadas como solución distribuida para la monitorización y se han realizado informes al respecto enfocados a distintos aspectos del sistema tales como la forma de tomar los datos, la distribución de los agentes para monitorizar adecuadamente una estructura, entre otros.[2]

4.1 Selección de la arquitectura

No existe una arquitectura general que pueda ser utilizada para cualquier problema, por lo cual se escogieron algunas utilizadas para problemas de monitorización y se efectuó una comparación entre ellas con el fin de obtener aquella que pueda cumplir de forma más sencilla con las especificaciones de monitorización de salud estructural. La Tabla 4 muestra una comparación entre las arquitecturas ya presentadas basándose en su flexibilidad para que un agente realice parte de la función de otro u otros, la cantidad de agentes con diferentes

funciones ya que esto afecta el tiempo de procesamiento de las señales, así como la dificultad de programación de los mismos y también se tomó en cuenta el tipo de arquitectura ya que en un futuro se espera llevar la arquitectura adaptada a un sistema real, por lo que es necesaria una interacción con el entorno. Además de los factores previamente señalados, se tomó en cuenta la documentación disponible sobre el desarrollo de la arquitectura, así como una clara explicación de las funciones individuales de los agentes que la componen. Los índices de flexibilidad pueden ser bajos, medios o altos dependiendo de la capacidad de los agentes para adaptarse a nuevos roles, o para suplir las funciones de otros que dejan de funcionar. El número de agentes hace referencia a los diferentes tipos de agentes que existan en la arquitectura. Como en algunos casos no es fijo debido a la flexibilidad, se considera que al menos debe existir un valor fijo de mínimo dos agentes para que el sistema sea considerado una arquitectura.

El tipo de arquitectura hace referencia a la definición de la misma de acuerdo a la clasificación presentada por Norvig [7].

Tabla 4. Comparación de las diferentes arquitecturas

Arquitectura	Flexibilidad	Número de agentes	Tipo de arquitectura
ARCHON	Baja	4	Híbrida
JEINA	Baja	6	Híbrida
ORG	Media	Depende del desempeño esperado	Lógica
MAGMA	Alta	Depende de la cantidad de problemas simultáneos	Lógica
Propuesta por Yuan	Media	10	Híbrida

La propiedad que presenta más prioridad para seleccionar es el tipo de arquitectura ya que se necesita una de tipo híbrido que sea capaz de analizar datos extraídos de sensores externos al sistema y analizar su contenido. De acuerdo con esta restricción las arquitecturas ARCHON, JEINA y la propuesta por Yuan et al. presentan las mejores alternativas. Sin embargo, el uso previo y exitoso de la arquitectura de Yuan [28]. en una estructura civil y su mediana flexibilidad, la hacen más adecuada para ser adaptada a trabajar con la metodología CBR.

4.2 Adaptación de la arquitectura

Esta arquitectura presenta una solución muy particular al problema de monitorización de salud estructural, ya que procesa la información de múltiples sensores y la fusiona en un punto común que utiliza el modelo Blackboard. Esta particularidad en el sistema local permite que se realicen diagnósticos independientes acerca de un mismo punto, y, gracias al modelo pizarra, el primer subsistema que presenta el diagnóstico influye en el diagnóstico de los demás. Tras realizar esta fusión local, la información pasa a un coordinador central encargado de obtener los diferentes datos de los subsistemas y organizar la ejecución de acciones por parte de los sistemas centrales. Este agente también se encarga de mostrar la información pertinente al usuario. Sin embargo, los agentes locales no se pueden comunicar con los agentes de fusión de otras localidades y por tanto no pueden suplir su función instantáneamente, por lo cual se espera que los coordinadores de comunicaciones locales se comuniquen entre sí para poder determinar fallas en la transmisión de datos. Debido a estas propiedades se requiere ajustar la arquitectura para que la información pase entre diferentes niveles locales, perdiendo un poco de independencia pero asegurando un mejor control de errores. Dado que se busca minimizar la cantidad de agentes para reducir la cantidad de hardware en una futura implementación, se observó que no es necesario tener un agente de coordinación y uno de comunicación, cuando

ambas funciones pueden ser ejecutadas por uno solo que maneje las informaciones disponibles así como los destinos de las mismas.

Los agentes de estimación local dependen una adquisición de una buena cantidad de información de la estructura en ese punto, porque si no sus diagnósticos locales son inútiles. Por tanto, estos agentes deben presentar una opción adicional que identifique si existen daños o no y solo envíe un mensaje corto, en vez de enviar la trama de datos y ejecutar un proceso innecesario. La modificación hecha a la arquitectura se puede observar en la Figura 8

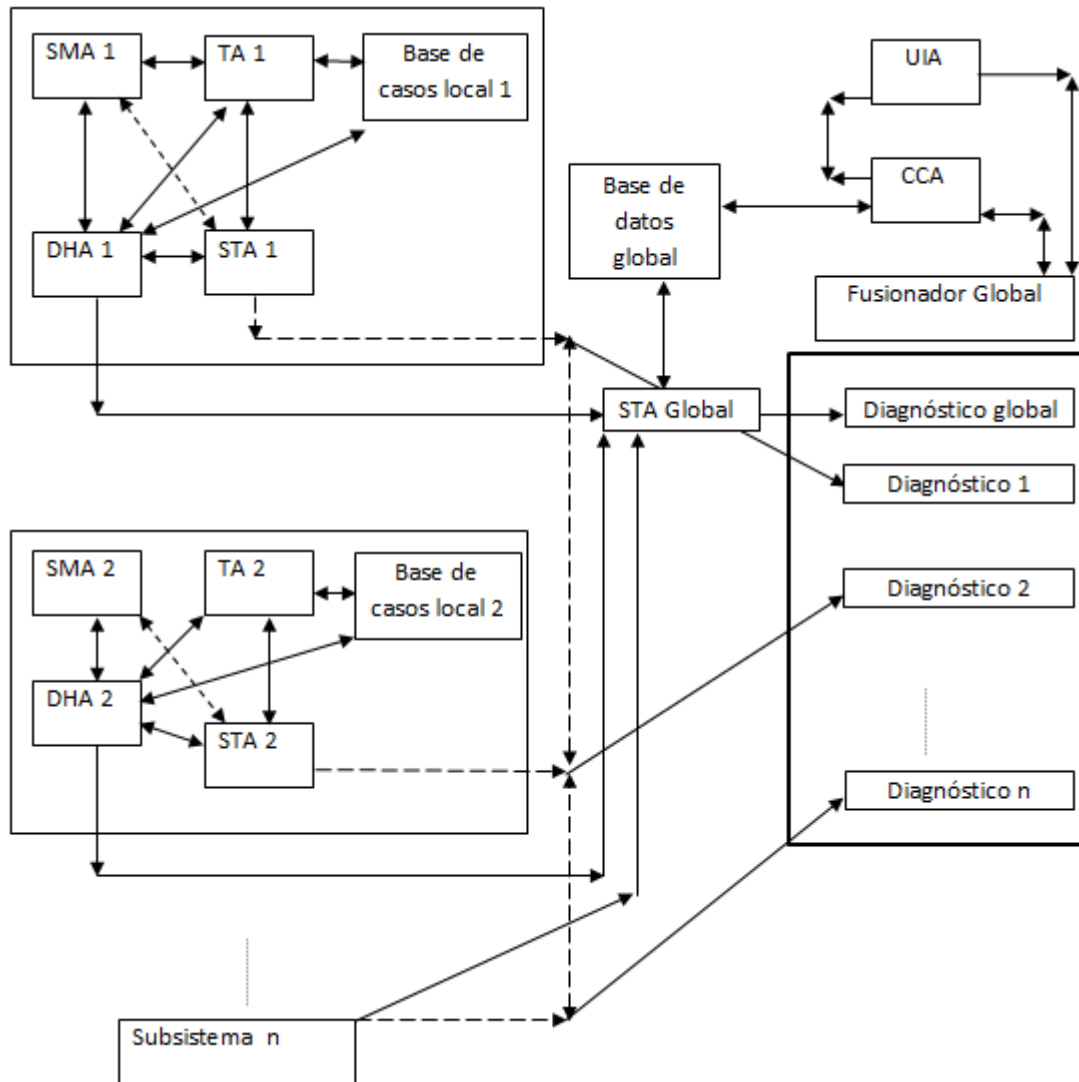


Figura 8 Adaptación de la arquitectura de Yuan et al.

Las líneas completas indican comunicaciones realizadas para efectuar el proceso de monitorización. Las líneas punteadas son comunicaciones de control, realizadas para observar el estado de un agente en un momento determinado.

De acuerdo con su funcionalidad se tienen siete tipos diferentes de agentes mientras que en la arquitectura original se tenían 10. Dada la reducción de funciones individuales de cada agente, así como del tipo de sensores se puede asumir que el sistema funcione un poco más rápido, aunque no hay manera de comparar los tiempos de respuesta.

La funcionalidad de estos agentes adaptados se describe a continuación:

Agente de monitorización estructural (SMA): Estos son los encargados de transformar los datos de los sensores al protocolo ACL. Su característica principal es la capacidad de detectar fallas en la adquisición de datos y reportarlo como un mensaje de tipo Inform, donde describe la falla. Su salida son los datos de aceleración hacia el DHA.

Agentes habilitadores de datos (DHA): Estos agentes maneja la información entre los diferentes puntos de la estructura, por lo que maneja la entrega de datos entre el sistema local y el sistema global. Dado que la herramienta JADE presenta un Habilitador al ejecutarse, no se construyó uno nuevo para suplir esta necesidad, debido a lo cual existieron ciertos inconvenientes en la transmisión de información con otros agentes habilitadores. La meta de este agente es llevar los mensajes a su destino con todo su contenido en el menor tiempo posible.

Agente traductor (TA): Los traductores convierten los datos obtenidos del SMA en información útil para los STAs. Este agente efectúa la correlación entre los datos del SMA local y los datos del SMA de referencia específico. Luego obtiene los modos de vibración utilizando transformada rápida de Fourier, obtiene las posiciones de los máximos locales en frecuencia y sus amplitudes, con lo que forma el vector de características que transmite al STA local.

Agente de transporte de la información de monitorización estructural (STA):

Este agente recibe el vector de características del nuevo caso y se comunica con la base de casos local para realizar la recuperación del caso con mayores similitudes. Además está pendiente de observar las respuestas de los demás agentes de su tipo en otras locaciones para advertir en caso de algún fallo con una bandera. Dado que presenta el diagnóstico local, es el encargado de enviarlo al DHA más cercano para que lo envíe al fusionador de datos global. El STA global obtiene información de datos correlacionados de toda la estructura para generar un diagnóstico local basado en la comparación con los datos presentes en la base de casos global, y este diagnóstico posee un mayor peso al ser evaluado por el fusionador debido a la cantidad de información con la que fue construido.

Interfaz de usuario (UIA): Provee la información relacionada con los diagnósticos locales y el diagnóstico global, así como el final, con el objetivo que el usuario tenga todos los datos disponibles y pueda juzgar la efectividad del sistema.

Agente de colaboración central (CCA): Es el centro de toda la operación, aunque si falla esto no impide que los diagnósticos locales se realicen. Este agente obtiene su información del fusionador y aplica las normas matemáticas explicadas en el capítulo anterior para determinar la dimensión global del daño, así como su severidad. También es el encargado de entregar a la interfaz de usuario los datos de diagnóstico definitivos, incluyendo la probabilidad de ocurrencia de cada daño.

Fusionador de datos: Este subsistema recibe todos los diagnósticos y organiza los elementos dañados, la cantidad de veces que cada uno aparece y la probabilidad de ocurrencia de daño, así como la severidad estimada por cada subsistema local. Su función es facilitar la tarea que cumple el CCA permitiéndole realizar búsquedas de forma más sencilla.

Cada uno de los componentes de la arquitectura realiza su función independientemente del desempeño de los demás, alertando al sistema central cuando no se encuentra la información esperada. El sistema puede realizar monitorización continua, pero la entrega del diagnóstico general a la interfaz de usuario solo se realiza cuando existe un operario humano que lo solicita.

El flujo de información entre los agentes para el proceso de monitorización se realiza de la siguiente manera:

- 1 Los SMAs reciben la información de la estructura, la convierten al protocolo ACL y la envían a los DHAs.
- 2 Los DHAs se comunican entre sí, analizan la cantidad de puntos de sensado y toman uno como referencia para que se realice la correlación entre los datos. Para este proyecto se trabajó con una referencia fija, pero esta puede variar tras una negociación entre los agentes.
- 3 Los DHAs envían los datos locales y de referencia a los TAs locales.
- 4 Los TAs locales crean los vectores de características a partir de su análisis en frecuencia, que luego devuelven al DHA respectivo.
- 5 Se envía el vector de características a los STAs locales y también al STA global para que se pueda realizar un diagnóstico general.
- 6 Los STAs ejecutan su función y presentan sus diagnósticos locales y el global, tras lo cual envían los datos al DHA respectivo que los entrega al fusionador global, el cual ya posee el diagnóstico global.

- 7 El CCA realiza su función y solo deja como resultado los elementos dañados con la severidad y dimensión calculadas, así como su probabilidad de ocurrencia. Como conoce la cantidad de diagnósticos que deben existir de acuerdo a la cantidad de puntos de sensado, realiza una evaluación corta para observar si su resultado tiene alta confiabilidad, o si al fallar algún sensor bajo la confianza sobre el diagnóstico final. Estas variaciones de confiabilidad son determinadas por la estructura específica y la cantidad de puntos de sensado.

- 8 El CCA envía los resultados a la interfaz de usuario, al tiempo que los diagnósticos locales y globales son mostrados. Cabe resaltar que cada diagnóstico esta diferenciado por el nombre del punto al cual pertenece con el fin de evitar errores de interpretación por parte del usuario.

VALIDACIÓN NUMÉRICA

Para realizar la validación del sistema multi-agente existen diversas formas desde las pruebas con ambientes controlados, hasta programas en capacidad de controlar todas las funciones de cada agente y variar sus ontologías acorde con la prueba. [30] [31]

Sin embargo se decidió utilizar un modelo numérico para poder validar el algoritmo de monitorización de salud estructural con un problema Benchmark descrito por el grupo en monitorización de salud estructural ASCE (American Society of Civil Engineering), el que posee modelos implementados en MATLAB® y permite generar la respuesta dinámica de la estructura metálica de la Universidad de British Columbia (UBC).

Para el diseño de una red multi-agente se presentan varias opciones como JADE [32] o la técnica de diseño de Prometeo (PDT) [33]. Sin embargo se observó que la herramienta JADE presentaba una documentación más detallada que las demás, su foro para las inquietudes de programación es muy completo y es actualizado constantemente por los creadores de la herramienta.

5.1 Caso numérico: Estructura Benchmark UBC.

Se utilizó la estructura metálica usada por Camacho [5] que presenta las frecuencias naturales mostradas en la Tabla 5.

Tabla 5. Frecuencias naturales de la estructura metálica.

Piso N°	Dirección X (Hz)	Dirección Y (Hz)
1	8.53	8.2
2	24.24	22.54
3	39.73	35.58
4	55.16	48.37

La respuesta dinámica fue simulada mediante 4 registros de aceleración por piso, dos en dirección X y dos en dirección Y. La señal de perturbación, denominada la excitación de este sistema es una fuerza horizontal aplicada en cada piso de la estructura modelando la fuerza del viento y se asume que es ruido gaussiano, el cual presenta componentes en todas las frecuencias, incluyendo las frecuencias naturales de la estructura. Según [34] y [5] para este Benchmark se considera la identificación de los siguientes patrones de daño:

1. Estructura sin daño.
2. Todos los refuerzos (brazos) del primer piso son removidos.
3. Todos los refuerzos (brazos) del primer y tercer piso son removidos.
4. Un refuerzo (brazo) del primer piso es removido.
5. Un refuerzo (brazo) del primer y tercer piso es removido.
6. Un refuerzo (brazo) del primer y tercer piso es removido y además una viga del primer piso sufre un decremento en el valor de la rigidez.
7. Daño en un solo elemento (viga, refuerzo o columna) de la estructura, en cualquier piso.
8. Daño en más de un elemento (máximo 4) del mismo tipo (viga, refuerzo o columna) y localizado en el mismo piso.
9. Daño en más de un elemento (máximo 4) de cualquier tipo (viga, columna o refuerzo) y localizado en el mismo piso.

De acuerdo con los patrones 8 y 9, la falla de cuatro elementos se considera un daño severo en un piso, por lo tanto en este proyecto la cantidad de daños simultáneos en una zona se limitó a cuatro elementos. Cada elemento es representado por un número entre 1 y 116 de acuerdo a su posición dentro de la estructura.

5.2 Generación de casos

Para la estructura *Benchmark UBC* se generaron de manera aleatoria 100 casos sin daño, 300 con daño en un solo elemento (disminución de rigidez entre 10% y 50% en pasos de 10%), 5860 casos con daño simultáneo (máximo 4 elementos del mismo tipo con disminución de rigidez entre 10% y 50% en pasos de 10%), 548 casos con 100% de disminución en la rigidez (*Full-Damage*, máximo 3 elementos del mismo tipo con daño simultáneo) y 6 casos correspondientes a los definidos en el problema *Benchmark*; para un total de 6814 patrones de daño para la etapa de entrenamiento. Para la etapa de validación se generaron 1870 casos entre daños en un solo elemento (280), daños simultáneos (1360), *Full-Damage* (204), estructura sin daño (20) y 6 casos correspondientes a los definidos como datos experimentales.

Se tomaron en cuenta los siguientes aspectos para generar los casos apropiados

1. Los datos de aceleración fueron *sub-muestreados* a 125 Hz, frecuencia que contiene información sobre todos los modos de vibración traslacionales de la estructura.
2. Para el cálculo de las funciones de densidad espectral de potencia se aplicó el método *welch* utilizando ventanas de 1024 puntos con 75% de *overlap*. Esto maximiza el promediado de los puntos mediante registros cortos.
3. Se utilizaron registros de 150 segundos de longitud, para un total de 70 promedios en el cálculo de la densidad espectral de potencia.
4. Para reducir los efectos del *Leakage* se utilizó una ventana *Hanning*.
5. Como canal de referencia para el cálculo de la densidad espectral de potencia se utilizó la aceleración del techo. Este canal fue seleccionado porque todos los modos de vibración son observados en este punto.

Debido a las características de la excitación, solo se registran aceleraciones en la dirección Y. Por lo tanto al utilizar el método Welch en cada registro de aceleración independiente se obtienen los 4 modos de vibración claramente diferenciados. A partir de estos modos de vibración se toma una vecindad de 5 Hz alrededor de cada frecuencia natural del sistema, se obtienen las amplitudes y posiciones de los máximos locales para formar un vector de características de 8 posiciones para cada piso. Un ejemplo de una densidad espectral de potencia del piso 1 sin daño se puede observar en la Figura 9

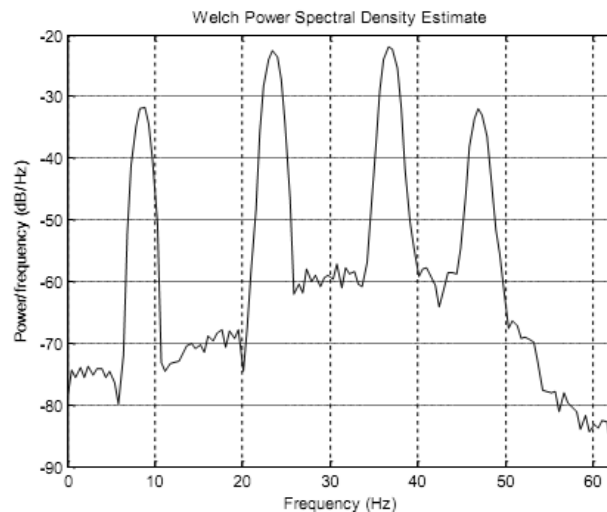


Figura 9 Densidad espectral de potencia de una aceleración en dirección Y

5.3 Identificación de daños

En esta sección se presentan los resultados del entrenamiento y validación. Se realizan cinco diagnósticos se buscan cinco errores de dimensión independientes, así como cinco errores de severidad, que luego se promedian para obtener un indicador de error más acertado. Para efectuar la validación cruzada se dividieron los casos en 3 grupos de igual tamaño (dos de 2271 y uno de 2272). Cada grupo cuenta con la misma cantidad de casos sin daño (33 y un grupo 34), casos de daño con un elemento (100), casos de daño con múltiples elementos (1953 y uno 1954), y casos de daño completo (182 y uno con 183).

La cantidad de falsos positivos y negativos en cada grupo y su porcentaje de aciertos se indican en la Tabla 6. Se tomó en cuenta el diagnóstico final para evaluar la cantidad de falsos positivos y negativos, ya que en algunas ocasiones un sensor indicaba que no existían daños pero al fusionar la información el sistema entregaba un diagnóstico con daño. Existen otros casos en los cuales solo un sensor indicaba un daño mientras que los demás indicaban un caso sin daño. Si el STA global no encontraba daño se asumió que era un error de percepción y se considera el diagnóstico final como estructura sin daño. El porcentaje de aciertos indica que tan efectivo es el sistema para encontrar daños en estructuras.

Tabla 6. Indicadores de falsos positivos y negativos con los grupos e validación cruzada

	Grupo de Validación 1		Grupo de Validación 2		Grupo de Validación 3	
	Falsos positivos	Falsos negativos	Falsos positivos	Falsos negativos	Falsos positivos	Falsos negativos
	2	40	2	42	1	32
Cantidad de aciertos	31/33	2231/2271	31/33	2230/2272	33/34	2239/2272
Porcentaje de aciertos	94%	98,23%	94%	98,15%	97,05%	98,6%

La cantidad de falsos negativos presentes es mucho mayor que la de falsos positivos debido a la cantidad de casos con daño en comparación con aquellos que no lo presentan. Se calcularon los errores de dimensión y piso para un grupo significativo de casos, que se muestran en la Figura 10 y la Figura 11.

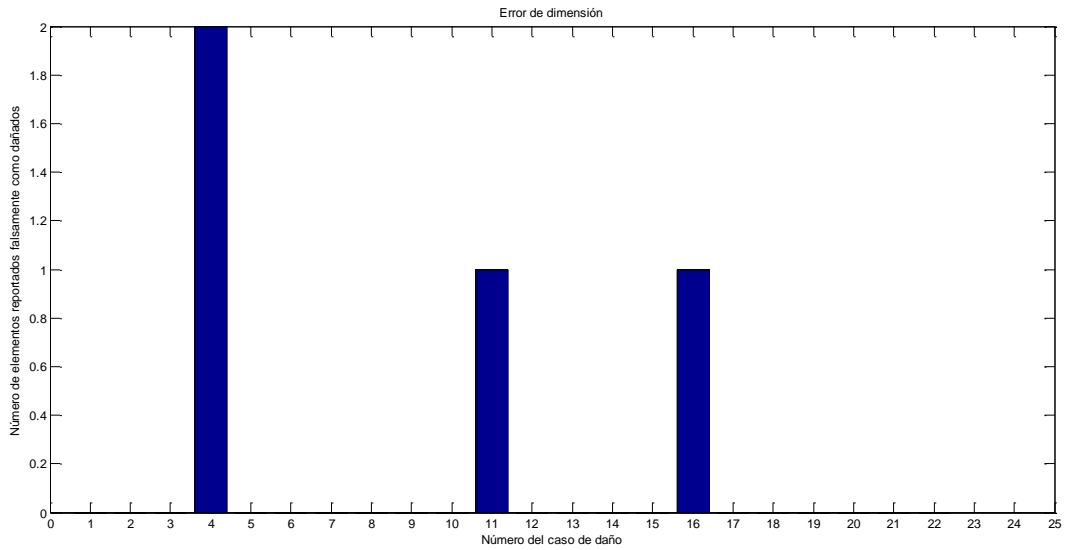


Figura 10 Error de dimensión

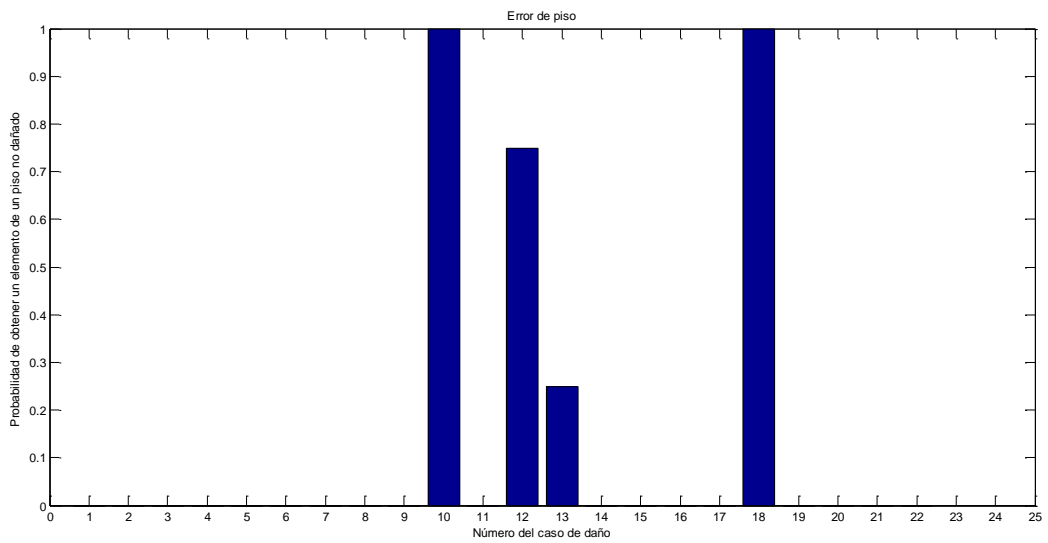


Figura 11 Error de piso

El error de severidad encontrado fue muy bajo en todos los casos debido a que la severidad afecta notablemente los modos de vibración y el sistema puede realizar un buen discernimiento de este parámetro.

5.4 Identificación con un sensor defectuoso

Se realizaron pruebas para determinar el error de detección por falta de información, para las cuales se utilizaron la totalidad de los casos de entrenamiento con el fin de observar el comportamiento del sistema cuando la información de uno de los 4 puntos específicos no es correcta. Las fallas en los sensores pueden ocurrir de diversas maneras y una de las más comunes ocurre cuando el sistema local envía un dato indicando que la estructura no posee daño (falso negativo local). Para tratar este tipo de errores, la metodología CBR utiliza la información de los demás sensores y realiza el diagnóstico final basado en la información adecuada, asumiendo que este diagnóstico local no es el indicado.

Debido a que la referencia del sistema es el sensor ubicado en el piso 4, se asume que este sensor presenta la mayor confiabilidad del sistema y que no presenta daños. Otro tipo de fallo ocurre cuando el sensor no envía ninguna información al sistema local y tampoco al global. En este caso se espera un tiempo de 5 minutos y se presenta un diagnóstico final basado en la información de los demás sensores, reemplazando el vector de características del sensor afectado por un vector que representa a la estructura sin daño con la fuerza de excitación fija que se utilizó para construir la base de casos. El tiempo de 5 minutos fue definido debido a la capacidad computacional de la máquina con la que se realizaron las pruebas, que entrega los resultados en menos de un minuto. Cuando se desee implementar en un sistema embebido se debe ajustar este parámetro de acuerdo a la velocidad de transmisión de datos, la latencia de los mismos y a la velocidad de procesamiento del sistema. Para determinar si el diagnóstico se encuentra en la zona local, se obtiene el error de piso descrito por [5]. Las pruebas se realizaron dañando un sensor a la vez, empezando por el sensor del piso 1 hasta llegar al ubicado en el piso 3 y los resultados se muestran en la Tabla 7. Se presentan problemas críticos de identificación de casos sin daño cuando más de un sensor deja de funcionar, lo cual dificulta una validación apropiada. [35]

Tabla 7 Promedio de errores con daños en los sensores

	Sensor # 1 dañado	Sensor #2 dañado	Sensor #3 dañado
Promedio del error de dimensión	23%	23%	25%
Promedio del error de severidad	37%	43%	34%
Promedio del error de piso	13%	21%	26%
Promedio de detecciones acertadas	81,25%	68,75%	62,5%

Existen casos donde se reporta error de piso al aplicar la metodología con todos los sensores funcionales. En esos casos el error de piso, generalmente, se mantuvo constante. Con excepción de los falsos positivos reportados anteriormente, todos los casos sin daño se detectan exitosamente cuando falla un sensor. El porcentaje de error de dimensión indica que, en promedio, cuando se daña un sensor el sistema presenta un elemento dañado por encima o por debajo de la cantidad correcta de daños. El porcentaje de error de piso indica que, en promedio, cuando se daña un sensor uno de los elementos en los que se identificó daño no pertenece a la zona dañada.

5.5 Aplicación de la metodología propuesta para un caso numérico específico

Se presenta un caso de daño donde se explica paso a paso la aplicación de la metodología. El caso evaluado presenta 4 elementos dañados en el segundo piso, con una severidad de 30% de daño.

Primero se utiliza el modelo de la estructura para simular las señales de aceleración de la estructura, recogidos por cada uno de los cuatro sensores. De esta manera los agentes de monitorización estructural SMAs obtienen la información de aceleración y la convierten a datos con el protocolo ACL. En caso que no encuentren información apropiada envían al DHA un mensaje indicando la falta de datos. Este proceso se describe en la Figura 12

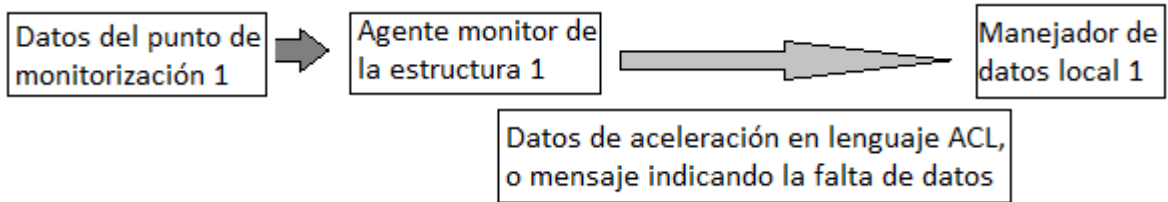


Figura 12 Paso 1 de la aplicación de la metodología a un caso numérico.

Luego el manejador de datos se comunica con otros manejadores, informando de la llegada de datos y obteniendo los datos del punto de referencia, que es la información del sensor #4 ubicado en el piso superior de la estructura. Este proceso se muestra en la

Figura 13. Las peticiones de información se realizan por medio de mensajes de tipo Query, y son respondidas por mensaje de tipo Inform.

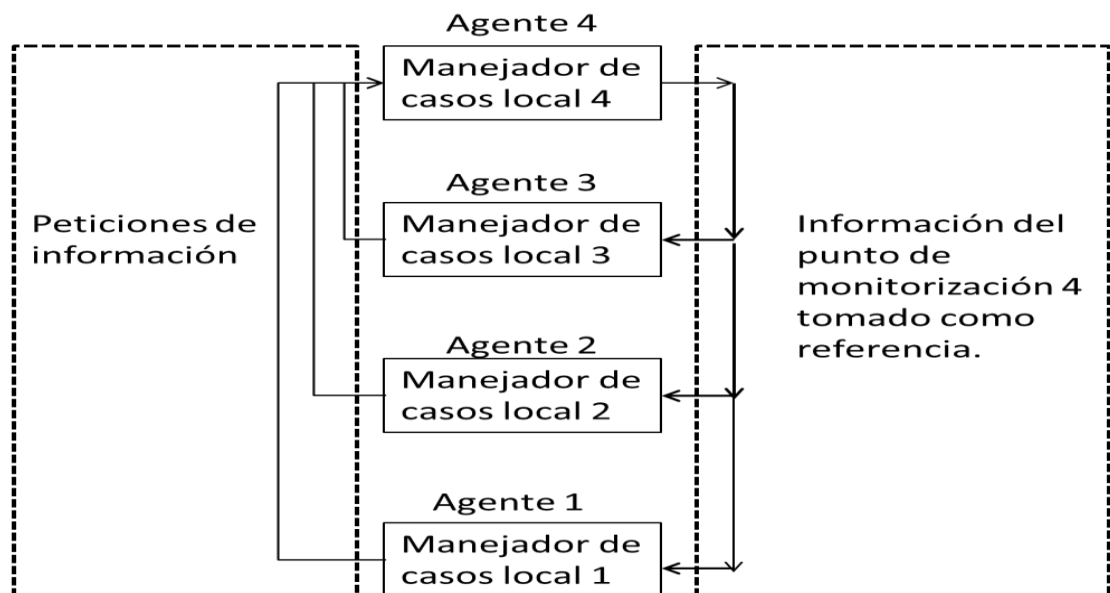


Figura 13 Paso 2 de la aplicación de la metodología a un caso numérico.

Tras recibir la información del piso de referencia el manejador envía los datos del monitor y los recibidos del manejador de casos 4 al agente traductor, el cual efectúa la correlación y el método Welch para crear y entregar un vector de características al manejador. Este proceso se realiza en los cuatro sub-sistemas locales y se muestra en la Figura 14

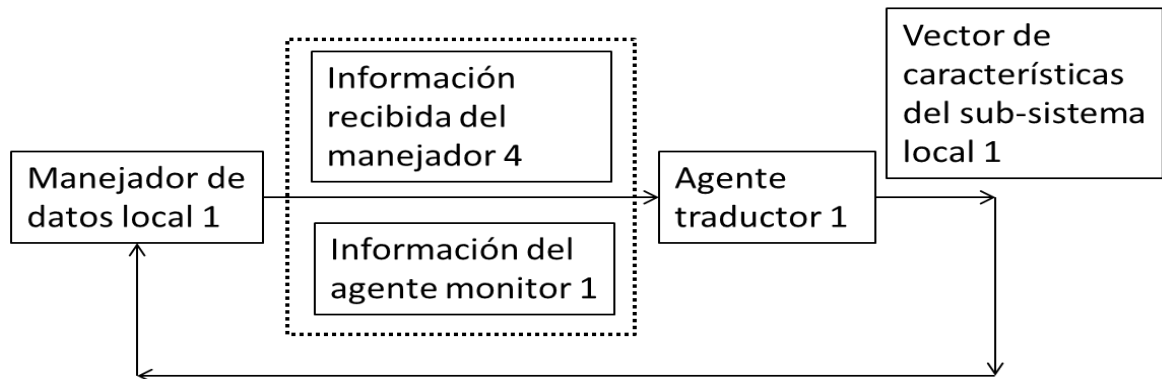


Figura 14 Paso 3 de la aplicación de la metodología a un caso numérico

Tras obtener el vector de características local, el manejador de datos envía la información al agente de transporte de la información de monitorización estructural (STA) local, y también lo envía al STA global. El STA local se comunica con la base de casos local, efectúa una búsqueda simple entre los vectores de características almacenados, calcula la distancia euclidiana con cada uno y obtiene los tres vectores con los que se obtuvo la menor distancia. Como los vectores recuperados indican diferentes casos de daño estructural se adquieren los diferentes elementos dañados y se observa cuantas veces se repite cada uno entre los diferentes vectores. De acuerdo a la distancia euclidiana de cada vector con el caso original, se calcula un factor de ocurrencia para cada elemento dependiendo del vector donde aparezca. Por último entrega un diagnóstico local indicando cuales son los elementos con mayor probabilidad de estar dañados en donde la cantidad de elementos dañados y su severidad está definida por las ecuaciones planteadas en el capítulo 3. Esta parte del proceso se presenta en la Figura 15 y se repite en todos los sub-sistemas que componen la arquitectura

multi-agente para obtener un total de 4 diagnósticos locales y que el STA global tenga 4 vectores de características en su entrada.

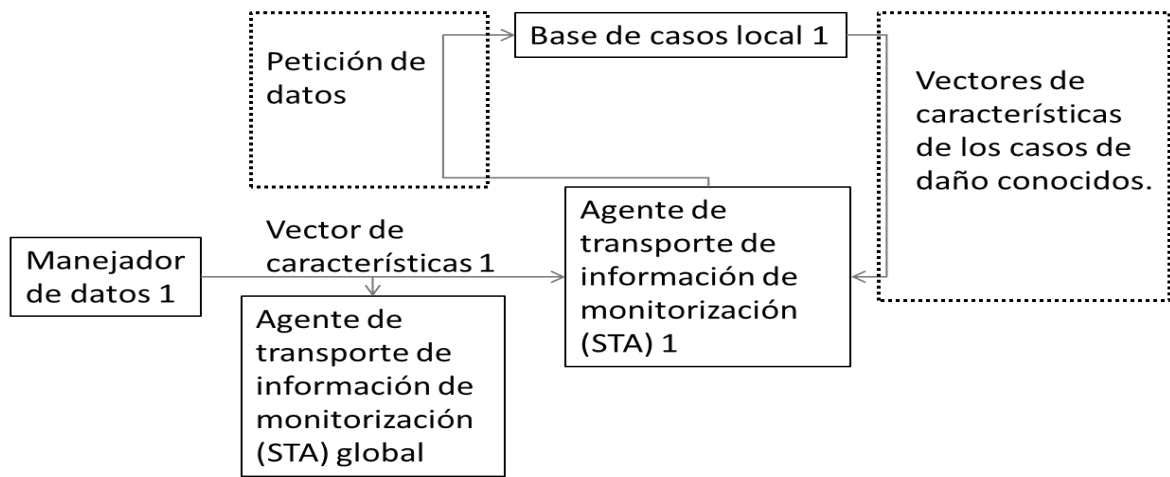


Figura 15 Paso 4 de la aplicación de la metodología a un caso numérico

El STA 1 realiza las operaciones mostradas en la Figura 16. Se describe el STA local 1, pero todos los demás STAs locales operan de igual manera. La respuesta obtenida de los STAs es un diagnóstico local compuesto por una estimación de la severidad del daño, los elementos en donde ocurrió y la probabilidad de que este daño sea correcto. El STA global recibe un vector de características de cada subsistema por lo que posee 4 vectores de entrada. Estos 4 vectores se concatenan para crear un vector de características global, que se compara con los vectores de 32 posiciones almacenados en la base de casos global. Este agente entrega un diagnóstico que presenta mayor credibilidad que aquellos obtenidos localmente ya que posee más información de toda la estructura para realizar su análisis. Su salida presenta los elementos dañados, la probabilidad de que cada uno de estos elementos esté realmente dañado y un estimado de la severidad del daño a nivel de toda la estructura.

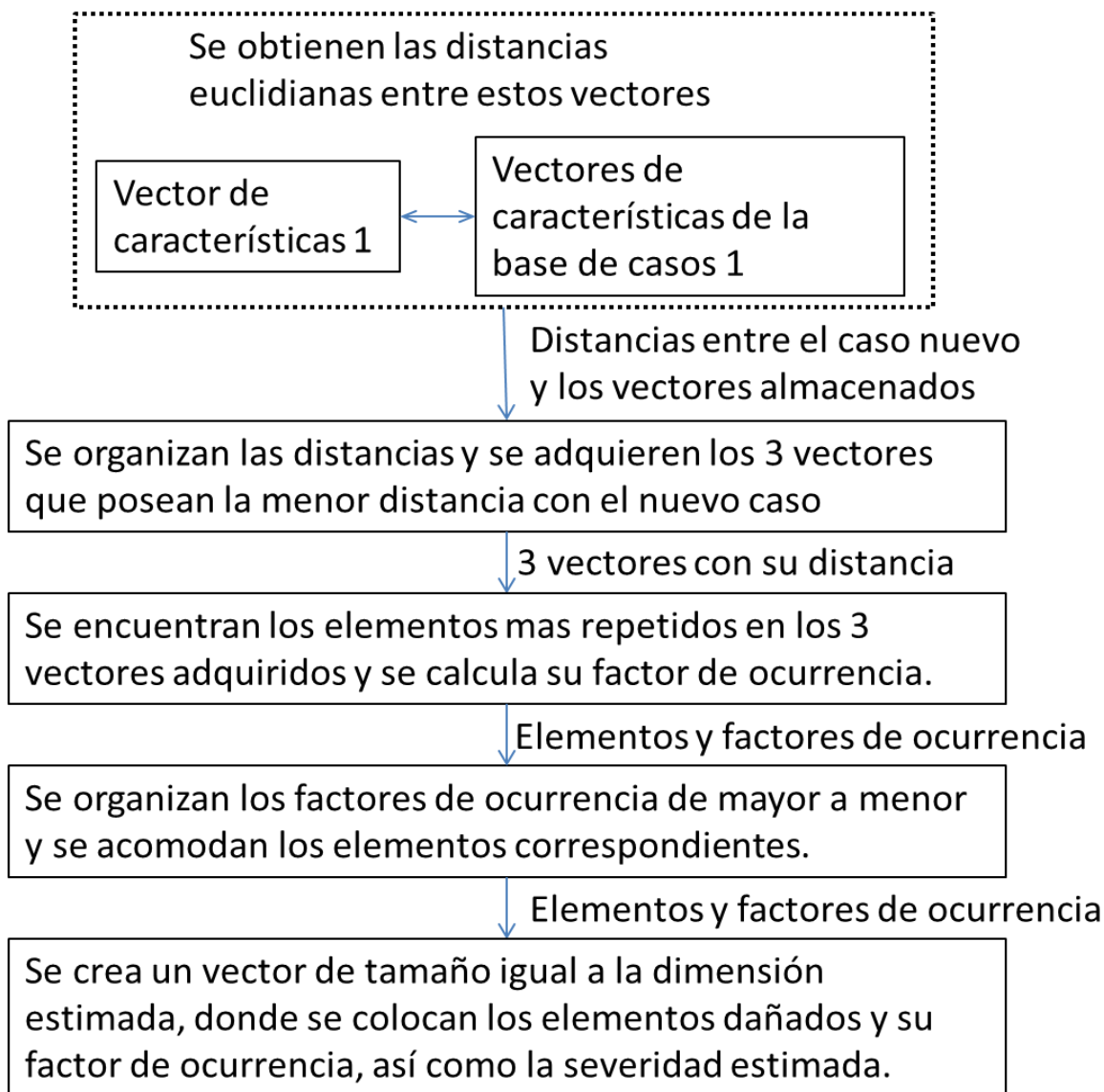


Figura 16 Operaciones realizadas en un agente de monitorización de estructura (STA)

Los diagnósticos locales y el diagnóstico del STA global se presentan en la Tabla 8. Las probabilidades de ocurrencia presentadas por el STA global son multiplicadas por 1,2, lo cual indica que tienen un 20% más de peso para elaborar el diagnóstico final. Esto se realiza basado en el supuesto que un diagnóstico global presenta un 20% más de confianza que uno local. Esta confiabilidad puede variar de acuerdo al número de STAs.

Tabla 8 Diagnósticos realizados por los STAs en un caso numérico

	Elementos dañados	Probabilidades de ocurrencia	Severidad del daño
STA 1	34-38-35-30	100 – 99 - 99 - 99	50%
STA 2	34-38-30-35	100 – 99 – 99 - 99	50%
STA 3	34-35-30-38	100 – 99 – 99 - 99	50%
STA 4	34-38-30-35	100 – 99 – 99 – 99	50%
STA global	32-33-34-36	120 - 120 - 120- 120	50%

Tras la obtención de los diagnósticos por parte de los STAs locales, estos devuelven a su manejador local respectivo un vector que contiene los elementos dañados, su probabilidad de ocurrencia y la severidad de daño estimada. El manejador envía estos datos a la interfaz de usuario y al fusionador de datos. Este elemento recibe los diferentes diagnósticos, estimaciones de severidad y organiza la información revisando cuales son los elementos que más se repiten en los diferentes diagnósticos y cuales presentan mayor probabilidad de ocurrencia, además de calcular una severidad general. Estos datos organizados son enviados al agente de colaboración central (CCA), el cual genera un diagnóstico final con un tamaño definido por la dimensión estimada y la severidad obtenida de ponderar las estimaciones de los STAs. El diagnóstico final se calcula sumando las probabilidades de ocurrencia de cada elemento, multiplicando por el número de veces que aparece el elemento en los 4 diagnósticos y multiplicándolo por 1,2 en caso que aparezca en el diagnóstico global, para luego dividir entre 5,2. Con esto se garantiza que los elementos que no se repitan probablemente correspondan a diagnósticos errados. Esta acción se muestra en la Figura 17.

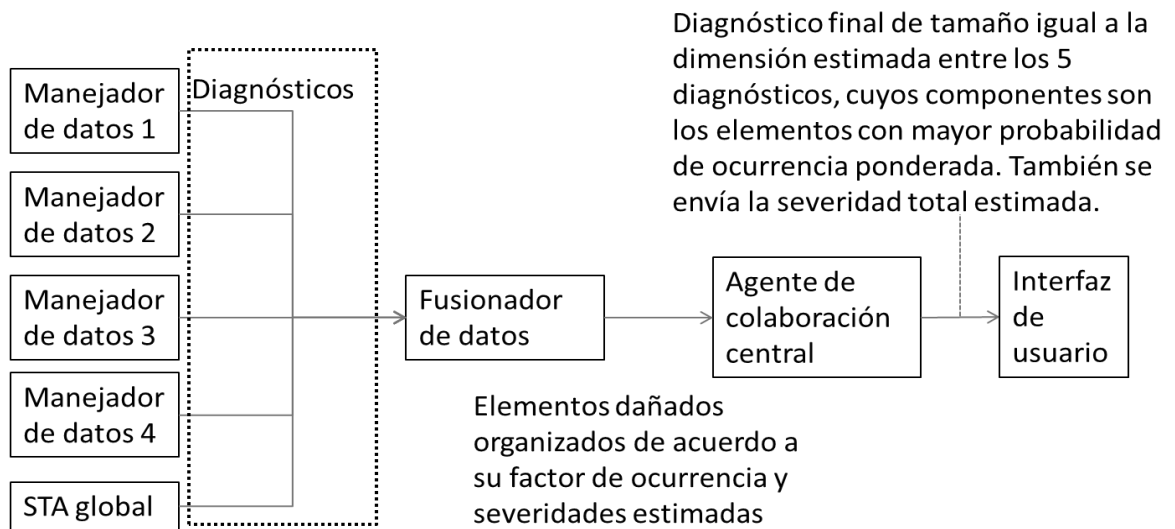


Figura 17 Paso 5 de la aplicación de la metodología a un caso numérico.

Los resultados de las operaciones realizadas en el fusionador y el agente de colaboración central se muestran en la Tabla 9. En este caso la dimensión estimada fue redondeada a 4 y la severidad estimada fue igual a 50% las cuales fueron calculadas de acuerdo a las ecuaciones presentadas en el capítulo 3

Tabla 9 Resultados obtenidos del fusionador de datos para un caso numérico.

Elementos con daño	Número de veces que se repite	Suma de la probabilidad de ocurrencia	Probabilidad de ocurrencia ponderada
34	5	520	100
38	4	396	76,15%
30	4	396	76,15%
35	4	396	76,15%
32	1	120	23,07%
33	1	120	23,07%
36	1	120	23,07%

La dimensión estimada es igual a 4, y por eso se toman los cuatro elementos con mayores probabilidades de ocurrencia. El diagnóstico final presentado al usuario se muestra en la Tabla 10 y los porcentajes de daño se muestran en la Figura 18

Tabla 10 Diagnóstico final del caso numérico

Elemento con daño	Probabilidad	Porcentaje de daño
34	100 %	50 %
38	76,92%	50 %
35	76,92%	50 %
30	76,92%	50 %

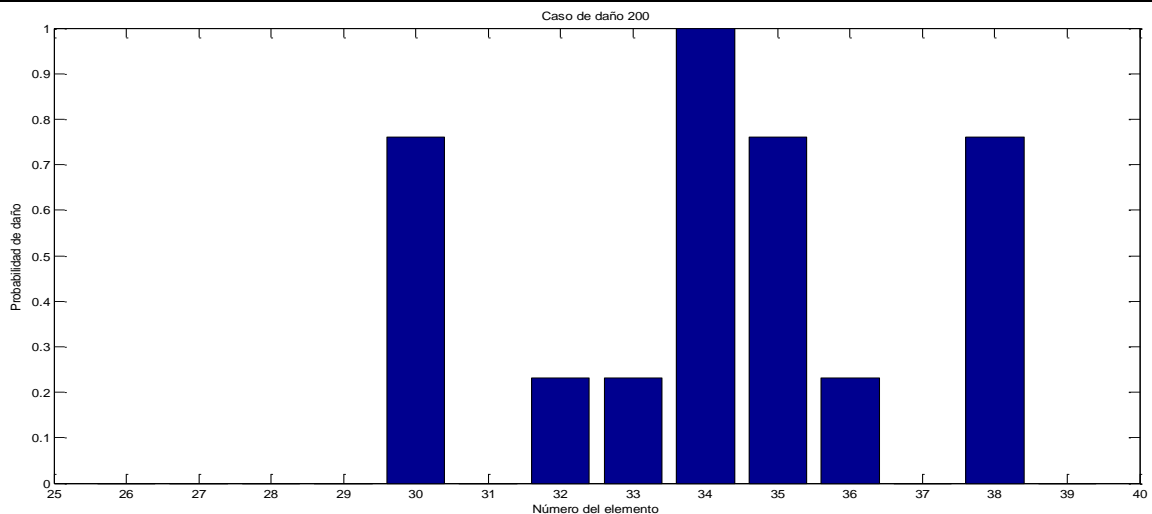


Figura 18 Porcentajes de daño de cada elemento identificado

Para este ejemplo el porcentaje de daño es el mismo para todos los elementos dañados, sin embargo existen casos donde la severidad general varía de acuerdo con los diagnósticos locales. Se consiguió realizar una detección del daño y en este caso se obtuvo correctamente el piso donde estaba ubicado. Sin embargo, debido al uso de los vectores de características, se encontraron múltiples casos donde no se identificaba correctamente el piso en donde ocurría el daño, o se presentaba un error en la dimensión del diagnóstico final.

CONCLUSIONES, RECOMENDACIONES Y APORTES

6.1 Conclusiones

La comunicación entre los agentes, impuesta por el protocolo ACL y la plataforma JADE, impide una mayor velocidad para calcular los diagnósticos porque su naturaleza Asíncrona no puede entregar los datos en el orden que son enviados, debido a lo cual se deben ejecutar algoritmos adicionales para organizar la información de llegada y determinar sus propiedades de origen.

La fusión de datos puede incurrir en múltiples errores de identificación de daño cuando no se tiene un algoritmo de ponderación, que determine cuál es el sub-sistema que presenta mayor interés para el obtener la información requerida por el usuario.

La descentralización de los datos permite obtener diagnósticos locales, en ocasiones muy aproximados al diagnóstico deseado, pero compromete el nivel de confiabilidad del sistema completo porque existen casos donde el diagnóstico solo se puede obtener uniendo toda la información, algo que no pueden realizar los subsistemas, por lo que el diagnóstico final presenta errores aun cuando un sistema si obtuvo la respuesta adecuada.

Una falla en el sensor de referencia limita la capacidad de procesamiento de todo el sistema, reduciendo la independencia de cada subsistema, sin embargo independizar completamente el sistema impide que se pueda utilizar la comparación de las respuestas espectrales al no poder separar el efecto de la fuerza de excitación de la respuesta natural del sistema.

El uso de agentes para monitorizar una estructura permite la detección de fallas en puntos estratégicos de la estructura, siempre y cuando los sensores estén correctamente posicionados, y la estimación de daños globales se puede realizar con muy poca información disponible, sin embargo su mayor limitante es la capacidad de computo que requiere el hardware para efectuar las operaciones indicadas por cada agente.

Cada subsistema, al estar localizado en un host individual, debe registrarse con los DFs de la plataforma JADE lo cual indica una pérdida de tiempo significativa a comparación del uso de un sistema experto con este mismo propósito.

Una implementación en hardware requeriría de un elemento como una FPGA o un sistema embebido para cada agente, especialmente para los encargados de realizar la búsqueda en las bases locales, ya que deben efectuar control de daños, búsqueda en memoria, comunicación y obtención de parámetros por medio de transformada de Fourier.

Varios casos de los pisos 2 y 3 fueron detectados pero no identificados correctamente, debido a que no afectaban de forma notable los modos de vibración y cuando se efectuaban las comparaciones con las bases de casos, se adquirían vectores no relacionados con el daño presente, afectando el diagnóstico final.

6.2 Aportes

Se adaptó una arquitectura de agentes inteligentes para el problema de monitorización de salud estructural.

Se utilizó la monitorización descentralizada en una parte del Benchmark ASCE

La identificación de la severidad del daño estuvo por encima del 80%, lo cual indica su efectividad en comparación con otros sistemas propuestos en el estado del arte.

Se obtuvo un algoritmo para la fusión de datos de aceleración provenientes de diversos sistemas

6.3 Recomendaciones

Dadas las limitantes impuestas por JADE, se puede utilizar otro lenguaje de programación que posea una comunicación sincronizada como GOAL para investigar cual es el mínimo tiempo necesario con un sistema computarizado para hallar un diagnóstico adecuados

Implementar las negociaciones entre agentes para definir el sensor más adecuado para que actúe como referencia del sistema, de acuerdo a los sensores disponibles y su información.

Utilizar elementos intérpretes más efectivos para poder definir el diagnóstico general de una forma más precisa. Dichos elementos pueden ser redes neuronales y algoritmos de reducción de características.

Debido a que los modos de vibración de este Benchmark particular son conocidos, se podría utilizar un algoritmo que solo requiera la información más relevante para cada modo de vibración particular. De esta forma se puede reducir el costo computacional del sistema.

Como la arquitectura es una adaptación de una ya existente, se recomienda la construcción de una arquitectura con la metodología GAIA que asegura el cumplimiento de metas, colocando como meta principal la velocidad de transmisión de datos correctamente.

La transformada wavelet puede ser incorporada dentro de la arquitectura para obtener los vectores de características que además de la información en el tiempo también presentaran información en el tiempo acerca del momento de inicio de la perturbación en el registro de aceleración con el fin de determinar el tiempo de duración de la misma.

BIBLIOGRAFÍA

- [1] C.R. Farrar and S. Doebling, "A Review of Structural Health Monitoring Literature : 1996-2001," *Structural Health Monitoring*, 2003, pp. 1996-2001.
- [2] N.G. Shrive, "Intelligent Structural Health Monitoring: a Civil Engineering Perspective," *2005 IEEE International Conference on Systems, Man and Cybernetics*, pp. 1973-1977.
- [3] P. Avitabile, "Experimental Modal Analysis (A Simple Non-Mathematical Presentation)," *Modal Analysis*, 2004, pp. 1-15.
- [4] M. Wooldridge, *MultiAgent Systems*, Liverpool: Jhon Wayley & Sons, 2002.
- [5] J. Camacho, "SISTEMA EXPERTO PARA LA MONITORIZACIÓN DE SALUD ESTRUCTURAL MEDIANTE EL RECONOCIMIENTO DE PATRONES: ADAPTACIÓN Y VALIDACIÓN NUMÉRICA," Universidad Industrial de Santander, 2010.
- [6] X. Zhao, S. Yuan, Z. Yu, W. Ye, and J. Cao, "Designing strategy for multi-agent system based large structural health monitoring," *Expert Systems with Applications*, vol. 34, 2008, pp. 1154-1168.
- [7] P. Norvig, "Intelligent agents," *Artificial Intelligence: A modern Approach*, Prentice Hall, 1995.
- [8] S.E. Lander, "Issues in multiagent design systems," *IEEE Expert*, vol. 12, Mar. 1997, pp. 18-26.
- [9] H. Voos, M.A. Lopez-Carmona, M.-M. Iván, and J.R. Velasco, *Multiagent Systems*, In-Teh, 2009.
- [10] M. Scheutz and V. Andronache, *The APOC Framework for the Comparison of Agent Architectures*, Notre Dame: 2004.
- [11] N.R. Jennings and W. College, "Applications of Intelligent Agents," *Intelligent agents*, 1996.
- [12] W. Jacak, K. Pro, and S. Dreiseitl, "Conflict Management in an Intelligent Multiagent Robotic System," *IEEE International Conference on Systems, Man, and Cybernetics*, Nashville: 2000, pp. 1793-1798.

- [13] H. Zhu, "A role-based architecture for intelligent agent systems," *Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, 2006.
- [14] S.Z.H. Zaidi, S.S.R. Abidi, M. S., and Y.-N. Cheah, "ADMI: A Multi-Agent Architecture to Autonomously Generate Data Mining Services," *International conference on intelligent systems*, IEEE, 2004, pp. 273-279.
- [15] L. Kong and L. Xiao, "A Multi-Layered Control Architecture of Intelligent Agent," *IEEE International Conference on Control and Automation*, Guangzhou: 2007, pp. 1454-1458.
- [16] B. Horling, R. Mailler, and V. Lesser, "A Case Study of Organizational Effects in a Distributed Sensor Network," *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, 2004, pp. 51-57.
- [17] W.R. Otte, J.S. Kinnebrew, D.C. Schmidt, and G. Biswas, "A flexible infrastructure for distributed deployment in adaptive sensor webs," *2009 IEEE Aerospace conference*, Mar. 2009, pp. 1-12.
- [18] A. Kazandzhiev, I. Momtchev, L. Popova, and D. Shikalanov, "Distributed multi-agent based approaches," *International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, Waltham: IEEE, 2005, pp. 3-8.
- [19] S. Ibarra, C. Quintero, J. De La Rosa, and J. R. Castan, "An Approach based on New Coordination Mechanisms to Improve the Teamwork of Cooperative Intelligent Agents," *2006 Seventh Mexican International Conference on Computer Science*, Sep. 2006, pp. 164-172.
- [20] T. Fukuda, I. Takagawa, and Y. Hasegawa, "From intelligent robot to multi-agent robotic system self-organizing robotic systems," *International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, Boston: 2003, pp. 413-417.
- [21] M. Marcela, P. David, and J.A.G. Luna, "Simulación basada en agentes inteligentes del proceso de negociación del mercado de energía eléctrica en Colombia," *Energética*, 2005, pp. 25-36.
- [22] M. Hedley, "Communication protocols for a structural health monitoring sensor network," *2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE Cat. No.04EX975)*, 2004, pp. 531-533.

- [23] N.R. Jennings, J.M. Corera, and I. Laresgoiti, "Developing Industrial Multi-agent Systems," *First international conference on multiagent systems*, 1995, pp. 423-430.
- [24] V. Lesser and D. Corkill, *History and Accomplishments of the Multi-Agent Systems Lab at the University of Massachusetts Amherst The Early Years*, Massachusetts: 2011.
- [25] M. Milano and A. Roli, "MAGMA: a multiagent architecture for metaheuristics.," *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 34, Apr. 2004, pp. 925-41.
- [26] M. Wooldridge, N.R. Jennings, and D. Kinny, "The Gaia methodology for agent-oriented analysis and design," *Proceedings of the third annual conference on Autonomous Agents*, 1999, pp. 69-76.
- [27] H.S. Nwana, D.T. Ndumu, L.C. Lee, B.T. Laboratories, and M. Heath, *ZEUS: An Advanced Tool-Kit for Engineering Distributed Multi-Agent Systems*, Ipswich, Suffolk: 2006.
- [28] S. Yuan, X. Lai, X. Zhao, X. Xu, and L. Zhang, "Distributed structural health monitoring system based on smart wireless sensor and multi-agent technology," *Smart Materials and Structures*, vol. 15, 2006, pp. 1-8.
- [29] A.F. Quintero-Parra and R. Villamizar-Mejía, "Estado del arte en monitorización de salud estructural: un enfoque basado en agentes inteligentes," *Ciencia e Ingeniería neogranadina*, vol. 20, 2010, pp. 1-14.
- [30] F. Klügl, "A validation methodology for agent-based simulations," *Proceedings of the 2008 ACM symposium on Applied computing*, New York, New York, USA: ACM Press, 2008, p. 39.
- [31] R. Fuentes, J.J. Gómez-sanz, and J. Pavón, "Verification and Validation Techniques for Multi- Agent Systems," *Upgrade*, vol. 5, 2004, pp. 15-19.
- [32] D.G. Fabio Bellifemine, Giovanni Caire, *Developing Multi Agent Systems with JADE*, John Wiley & Sons, 2007.
- [33] J. Thangarajah and M. Winikoff, "Tool Support for Agent Development using the Prometheus Methodology," *Fifth International Conference on Quality Software*, leee, 2005, pp. 383-388.

- [34] Juan Martin Caicedo, "Structural Health Monitoring of Flexible Civil Structures," Washington University, 2003.
- [35] J. Wu, S. Yuan, S. Ji, G. Zhou, Y. Wang, and Z. Wang, "Multi-agent system design and evaluation for collaborative wireless sensor network in large structure health monitoring," *Expert Systems with Applications*, vol. 37, Mar. 2010, pp. 2028-2036.

ANEXOS

ANEXO A. TEORÍA DE AGENTES INTELIGENTES

El término hace referencia a un programa que no actúa como un objeto reactivo, sino a uno que observa su entorno constantemente para poder actuar cuando sea necesario. Este tipo de programa representa una aproximación al comportamiento humano ya que basa su funcionamiento en proponer acciones de acuerdo a lo que percibe del entorno y a sus capacidades individuales y este comportamiento está basado en un modelo de funcionamiento conocido como BDI (Belief Desire Intention) (ver Figura 19) que muestra la manera en que una creencia afecta la toma de decisiones de un programa.

Una característica importante de los agentes es su capacidad de comunicación, ya que de acuerdo al tipo de mensaje, el mensaje con el cual el agente responde y la acción ejecutada pueden ser diferentes de acuerdo al ambiente percibido.

Sin embargo un elemento que reaccione ante un estímulo con una acción predeterminada puede ser considerado también como un agente inteligente mientras que pueda interactuar con otros elementos.

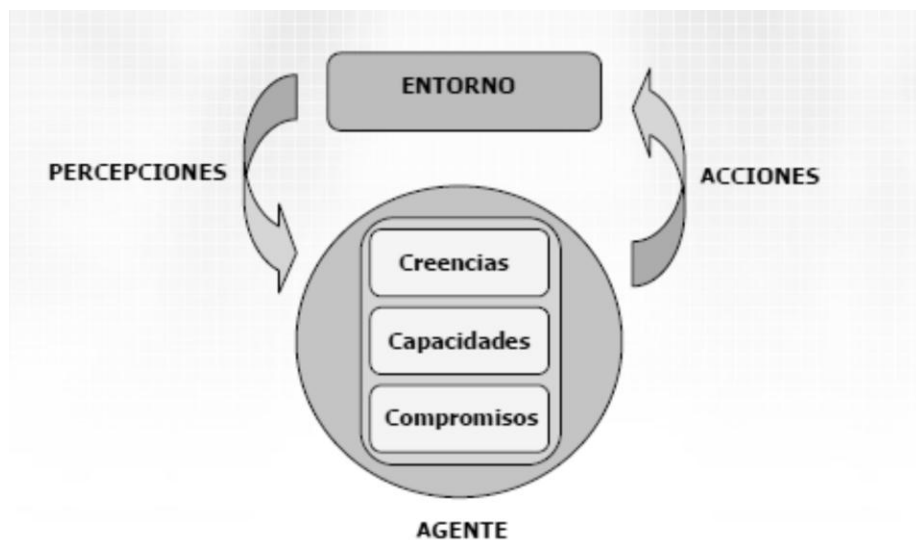


Figura 19 Modelo BDI de los agentes inteligentes

Adaptada de una presentación hecha por Eduard Muntaner, Esteve del Acebo, Josep Lluís de la Rosa

A.1 Tipos de agente

De acuerdo con los agentes inteligentes se pueden clasificar en cinco tipos de acuerdo a su capacidad y su grado de "inteligencia" percibida. Estos tipos serán descritos a continuación.

A.1.1 Agente de reflejo simple

Es un programa que reacciona de forma directa ante un estímulo de entrada. Normalmente se les conoce como objetos ya que solamente reaccionan ante un evento para entregar una o más salidas predeterminadas. Este tipo de agentes son muy utilizados en las redes de sensores, ya que son los encargados de proporcionar entradas al sistema en un lenguaje que éste entienda. Sus condiciones de acción no presentan algoritmos lógicos, los cuales generarían un nivel de inteligencia superior. De acuerdo con, cuando se implementan múltiples agentes de este tipo, la cantidad de mensajes enviados puede interferir con el funcionamiento general del sistema, pero estas eventualidades de comunicación pueden ser evitadas bajo una correcta programación de roles por parte de los demás agentes y un esquema organizado de tiempos límite entre comunicaciones para evitar cuellos de botella y demás problemas.

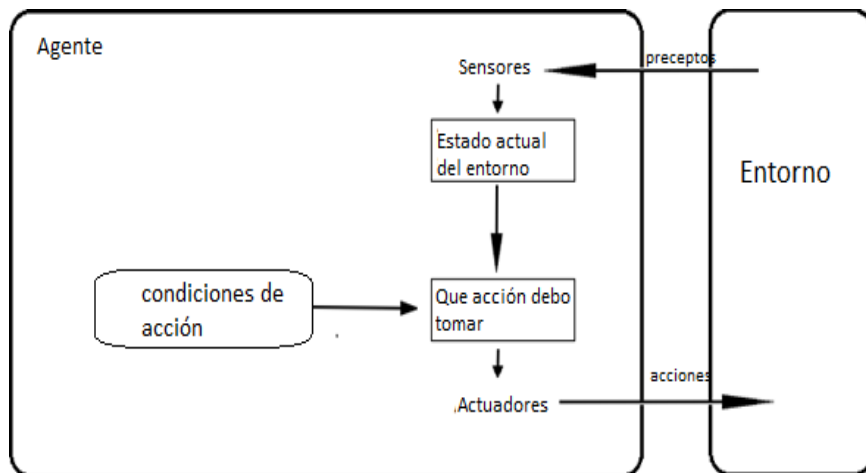


Figura 20 Esquema de comportamiento de un agente tipo reflejo simple

Adaptada de [7]

A.1.2 Agente de reflejo basado en un modelo

Es un elemento que reacciona ante una entrada aplicando un modelo matemático para entregar una o más salidas.

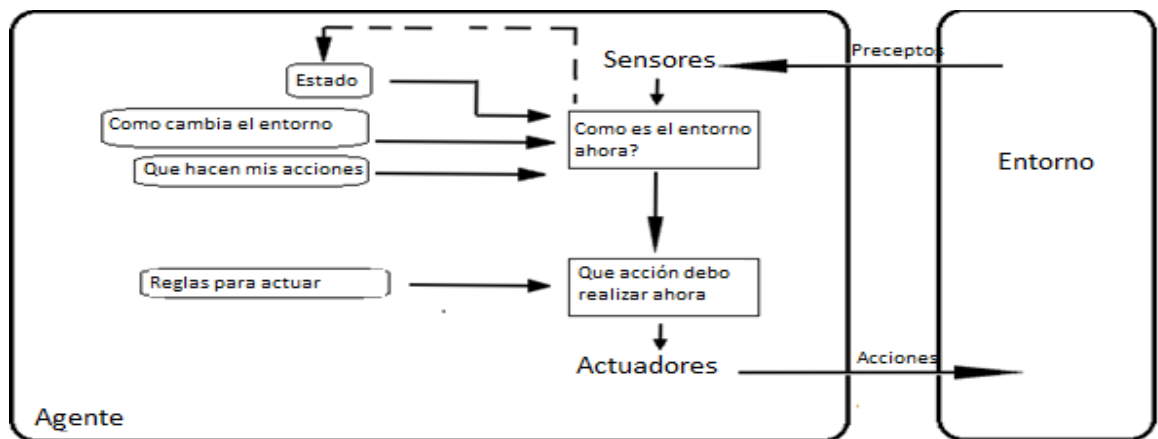


Figura 21 Esquema de comportamiento de un agente de reflejo basado en un modelo

Adaptada de [7]

A.1.3 Agente enfocado a una meta

Cuando se tiene programada una meta individual, el agente puede elegir entre varias opciones que lo lleven a cumplirla a corto o largo plazo. Este tipo de agentes generalmente poseen una función objetivo y varios algoritmos de optimización para encontrar la mejor respuesta sin ninguna restricción.

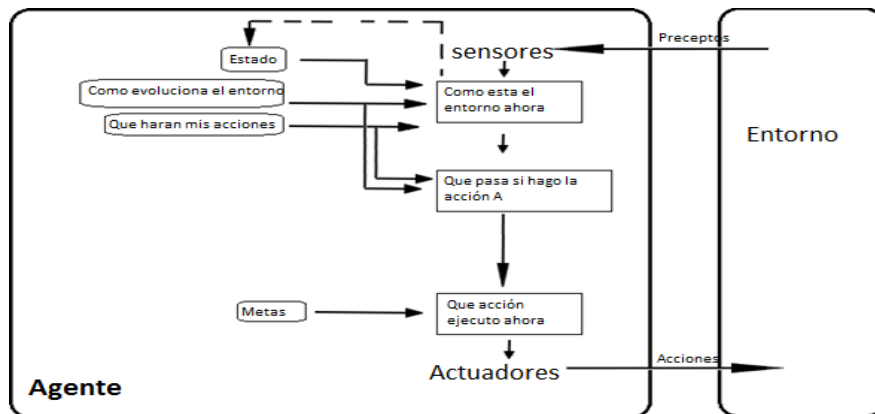


Figura 22 Esquema de operación de un agente enfocado a una meta

Adaptada de [7]

A.1.4 Agente basado en la utilidad:

Debido a que los agentes orientados a metas solo observan los estados que le lleven a esa meta y evitan todos los que no lo hagan, pueden llegar a un estado de suspensión, lo cual no es deseable muchas veces. Los agentes que tienen una función de utilidad pueden expresar una medida para determinar qué tan conformes o “felices” se sienten en un determinado estado que no esté incluido dentro de las metas a cumplir.

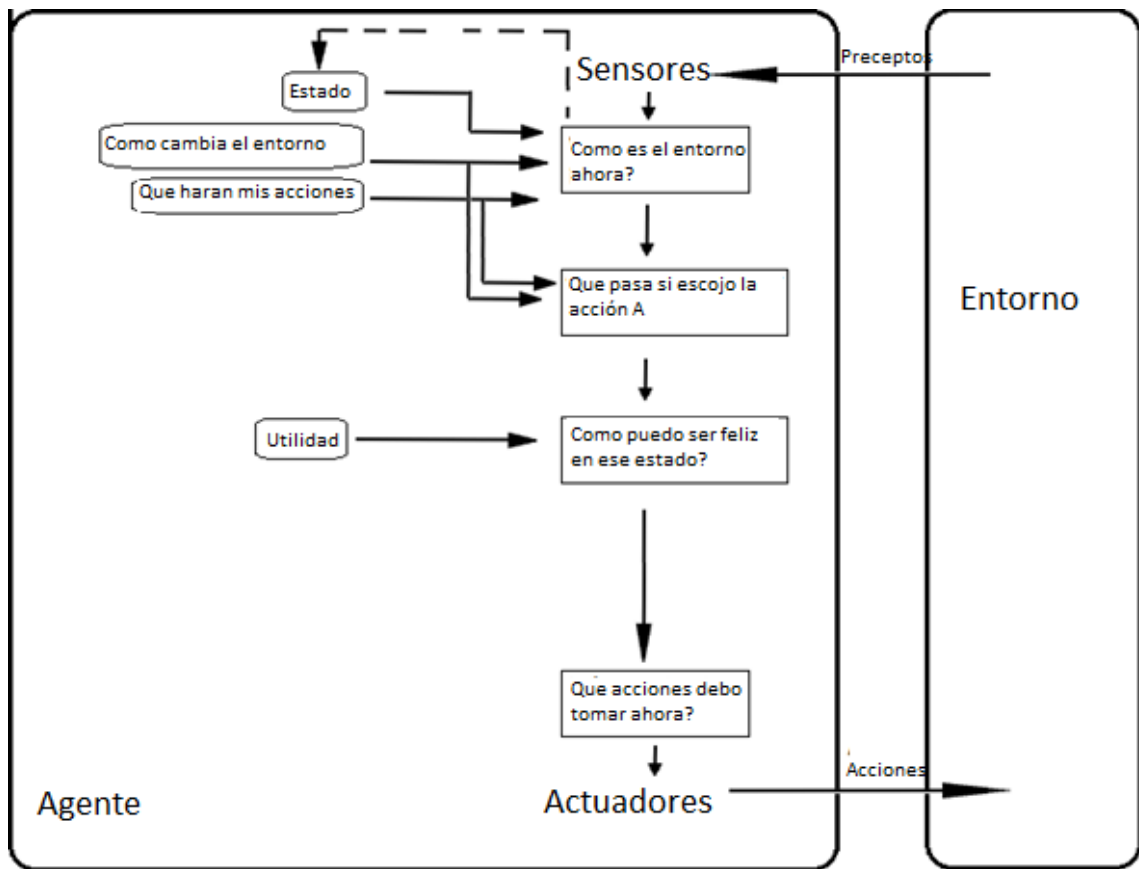


Figura 23 Esquema de operación de un agente basado en la utilidad

Adaptada de [7]

A.1.5 Agente de aprendizaje continuo

Estos programas presentan un comportamiento muy similar a un sistema experto. Poseen un crítico que indica el funcionamiento del agente en comparación con su estándar de comportamiento. También posee dos elementos, uno que aprende de las experiencias del desempeño del programa, para generar problemas acordes para poder lograr el aprendizaje deseado. Los problemas que se crean buscan enfrentar al programa con nuevas experiencias para poder desarrollar nuevas experiencias.

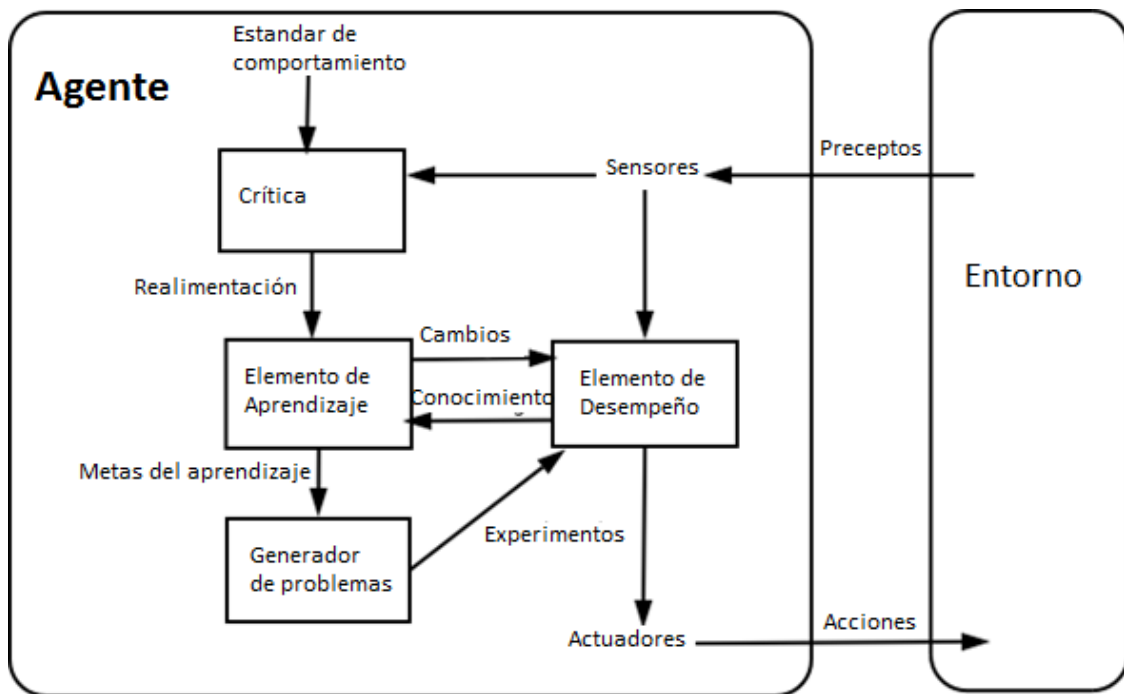


Figura 24 Esquema de operación de un agente de tipo aprendizaje continuo

Adaptada de [7]

A.2 Propiedades

Según y las principales propiedades de los agentes inteligentes son su capacidad de comunicación, su introspección, su proactividad, su capacidad de colaboración dependiendo del tipo de agente y su organización.

La proactividad se define como la capacidad de anticiparse a los hechos para proponer una solución adecuada, sin embargo esta es una propiedad muy difícil de observar en los sistemas ya que las respuestas han sido programadas previamente y el agente no puede “crear” una respuesta en el sentido estricto de la palabra. Usualmente el sistema reacciona de una forma determinada por el usuario cuando las condiciones del entorno sean adecuadas para este fin, y también puede observarse cuando un agente efectúa tareas de otro que no puede cumplirlas en ese momento.

La comunicación entre agentes puede presentarse como un simple paso de mensajes, hasta una negociación basada en modelos de toma de decisiones. Debido a los diferentes tipos de agentes es difícil utilizar un set de datos que sea entendido por todos los agentes de igual forma, por lo que se han desarrollado protocolos de comunicación exclusivos para agentes tales como el ACL (Agent Communication Language) desarrollado por FIPA(Foundation for Intelligent Physical Agents) o el KQML (Knowledge Query and Manipulation Language).

La ontología de cada agente es una expresión formal estandarizada que describe su función dentro de un sistema. Los componentes de la ontología son las clases, atributos, relaciones y reglas. Las clases son un grupo de elementos que representa un agente individual.

Las relaciones son las propiedades que se definen para interactuar entre diversos agentes y son determinadas por los tipos de agentes y sus capacidades ya que se puede definir una jerarquía en la que los de mayor rango tienen una relación de dominio con los demás.

Los atributos de cada clase son propiedades específicas que ayudan a diferenciar a varios agentes de la misma clase entre sí.

Las reglas y restricciones son componentes fundamentales de la ontología ya que definen las creencias individuales de cada agente y limitan la funcionalidad del mismo cambiando los estados objetivos a los que debe llegar el agente.

La introspección de un agente permite la toma de decisiones basada en sus capacidades y se realiza de diferentes formas de acuerdo al nivel de “inteligencia” percibida en el agente ya que los agentes de reflejo no son capaces de elaborar un diagnóstico de sus habilidades. La forma más sencilla de probar esta capacidad es alimentar el sistema con entradas de un tipo diferente al esperado y comprobar el tipo de salida.

El trabajo colaborativo es la unión de varios agentes para realizar una tarea conjunta que uno solo no pueda realizar. Esta decisión puede ser tomada por el agente de forma autónoma dependiendo de sus capacidades percibidas.

ANEXO B DESCRIPCIÓN DE JADE (JAVA AGENT DEVELOPMENT FRAMEWORK)

B.1 Descripción general de la plataforma Middleware

JADE es una plataforma middleware basada en JAVA, utilizada para el desarrollo de programas con agentes inteligentes. Utiliza el protocolo ACL (Agent Communication Language) para la comunicación entre sus diferentes partes desarrollado por FIPA (Foundation for Intelligent Physical Agents). Es un software de libre distribución desarrollado por TELECOM en Italia.

El programa crea múltiples contenedores para los agentes, cada uno de los cuales puede ejecutarse en uno o en varios sistemas. Un conjunto de contenedores constituye una plataforma. Cada plataforma debe tener un contenedor principal que tiene dos agentes especiales denominados AMS y DF

El AMS (Agent Management System) controla la plataforma. Es el único que puede crear y destruir a otros agentes, destruir contenedores y parar la plataforma.

El DF (Directory Facilitator) proporciona un directorio que anuncia qué agentes hay disponibles en la plataforma.

Dado que está basado en java, cualquier compilador de este lenguaje puede ser utilizado para programarlo. Existen extensiones que ayudan a la integración de la plataforma con el compilador Eclipse, pero Net Beans permite mayor facilidad de programación y depuración.

La relación de la plataforma JADE con otras plataformas basadas en el protocolo FIPA se presenta en la Figura 25.

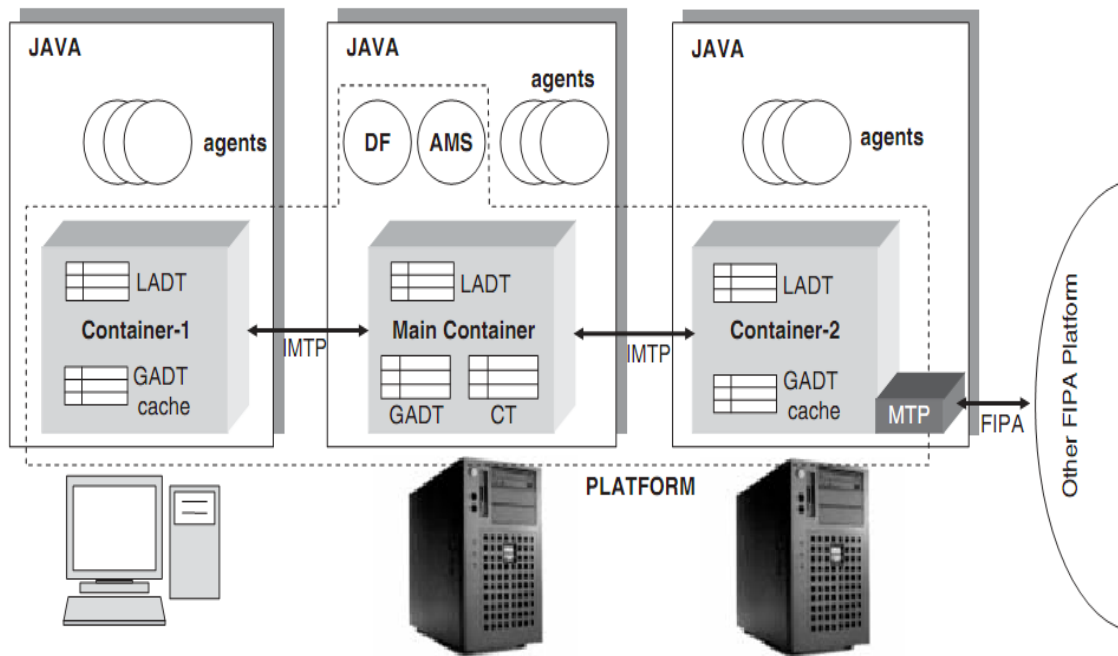


Figura 25 Relación de JADE con otras plataformas basadas en el protocolo FIPA

Tomada de [32]

Los elementos principales de la arquitectura JADE se relacionan de una forma jerárquica. Cada agente creado debe registrarse con el DF para poder estar ubicado dentro de una plataforma, aunque puede cambiar de plataforma si es un agente móvil. Tras identificarse ya se considera parte integral del sistema y puede ejecutar su meta local y acogerse a las metas globales indicadas por el programa principal de la plataforma. Cada plataforma posee metas independientes y equivalen, como se indica en la Figura 25 a sistemas distribuidos no centralizados.

El proceso de registro de un agente con una plataforma se indica en la Figura 26

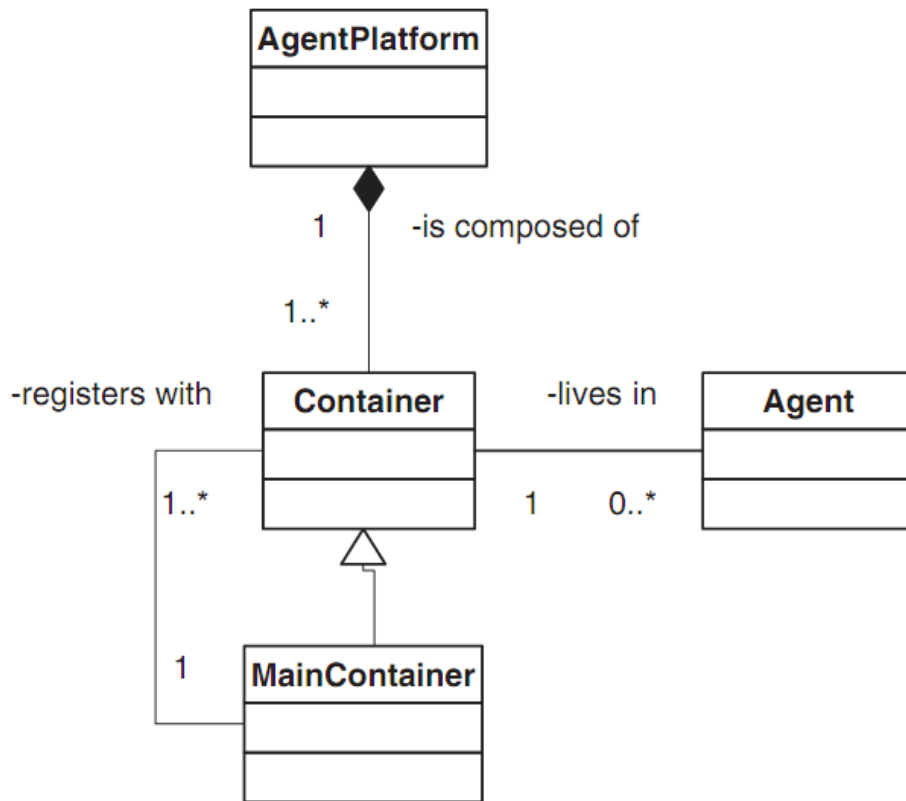


Figura 26 Forma de registro de los agentes con una plataforma específica

Tomada de: [32]

B.2 Mensajes en la plataforma JADE

Los agentes se comunican con diferentes tipos de mensajes propuestos por el protocolo ACL. Estos mensajes son los siguientes:

Accept Proposal: Es la acción de aceptar una propuesta hecha para ejecutar una acción.

Agree: La acción de estar de acuerdo para ejecutar una acción, posiblemente en el futuro.

Cancel: La acción de un agente informando a otro de que el primero no tiene intención de ejecutar cierta acción.

Call for Proposal: La acción de llamar para recibir proposiciones para ejecutar una acción dada

Confirm: El emisor informa al receptor que una proposición dada es verdadera cuando el receptor no tiene certeza acerca de la proposición.

Disconfirm: El emisor informa al receptor que una proposición dada es falsa, cuando el receptor cree que la proposición es verdadera.

Failure: La acción de decirle a otro agente que se intentó realizar la acción, pero que el intento fue fallido.

Inform: El emisor informa al receptor que una proposición es verdadera.

Inform If: Es una macro-acción para el agente, cuando debe informar al receptor si la proposición es verdadera o falsa.

Inform Ref: Es otra acción de nivel macro, que permite al emisor enviar al receptor un objeto que el emisor crea que es una información valiosa, ejemplo una fecha.

Not Understood: El emisor del acto informa al receptor que ha sentido cierta acción, pero no entendió que sucedió, el caso más particular ocurre cuando no se entiende el mensaje enviado.

Propagate: El emisor intenta enviar un mensaje embebido a un receptor, y que este al recibirlo, identifique las identificaciones de otros agentes y envíe el mensaje a quienes corresponda.

Propose: La acción de someter una propuesta a consideración para ejecutar una determinada acción dadas ciertas condiciones.

Proxy: El emisor desea que el receptor seleccione otros agentes denotados por una descripción ya enviada y que luego envíe un mensaje embebido a ellos.

Query: If: Es la acción de preguntarle a otro agente si una proposición es verdadera o falsa.

Query Ref: Es la acción de preguntarle a otro agente si el objeto referido por una expresión referencial.

Refuse Es la acción de rechazar la idea de ejecutar una acción, explicando la razón para el rechazo.

Reject Proposal: Es la acción de rechazar una propuesta para ejecutar una acción durante una negociación.

Request: El emisor pide que el receptor ejecute una acción.

Request When: El emisor desea que el receptor efectúe una acción bajo ciertas condiciones y luego que una proposición se vuelva verdadero.

Request Whenever: El emisor desea que el receptor efectúe una acción cada vez que cierta proposición se vuelva correcta.

Subscribe: Es la acción para que se requiera una intención persistente para notificar al emisor de un valor y cuando el objeto se modifique.