

Avances en Informática y Automática Octavo Workshop



**VNiVERSiDAD
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL



MÁSTER OFICIAL
EN SISTEMAS INTELIGENTES

Avances en Informática y Automática

Octavo Workshop

Avances en Informática y Automática

Octavo Workshop

Editores:

Alfonso González Briones

Isabel Martín Rodríguez

Publicado en España por:

Departamento de Informática y Automática
Facultad de Ciencias
Universidad de Salamanca
Plaza de los Caídos s/n
37008, Salamanca, España
Tel.: + 34 923294653
Fax: + 34 923294514
Web: <http://mastersi.usal.es>
Email: mastersi@usal.es

ISBN - 10 84-697-1520-8

ISBN - 13 978-84-697-1520-8

Editores:

Alfonso González Briones
Isabel Martín Rodríguez

Este obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 España.



Prólogo

El Máster Oficial en Sistemas Inteligentes de la Universidad de Salamanca tiene como principal objetivo promover la iniciación de los estudiantes en el ámbito de la investigación. El congreso organizado por el Departamento de Informática y Automática que se celebra dentro del Máster en Sistemas Inteligentes de la Universidad de Salamanca proporciona la oportunidad ideal para que sus estudiantes presenten los principales resultados de sus Trabajos de Fin de Máster y obtengan una realimentación del interés de los mismos.

La octava edición del *workshop* “Avances en Informática y Automática”, correspondiente al curso 2013 - 2014, ha sido un encuentro interdisciplinar donde se han presentado trabajos pertenecientes a un amplio abanico de líneas de investigación, desde los sistemas multiagente, la visualización de la información o la minería de datos hasta la lógica, el reconocimiento de patrones o las redes neuronales. Todos los trabajos han sido supervisados por investigadores de reconocido prestigio pertenecientes a la Universidad de Salamanca, proporcionando el marco idóneo para sentar las bases de una futura tesis doctoral. Entre los principales objetivos del congreso se encuentran:

- Ofrecer a los estudiantes un marco donde exponer sus primeros trabajos de investigación.
- Proporcionar a los participantes un foro donde discutir ideas y encontrar nuevas sugerencias de compañeros, investigadores y otros asistentes a la reunión.
- Permitir a cada estudiante una realimentación de los participantes sobre su trabajo y una orientación sobre las futuras direcciones de investigación.
- Contribuir al desarrollo del espíritu de colaboración en la investigación.

Organización

El *workshop* “Avances en Infomática y Automática” está organizado por el Departamento de Informática y Automática de la Universidad de Salamanca.

Program Committee

Francisco Javier Blanco Rodríguez
Belén Curto Diego
José Rafael García-Bermejo Giner
Luis Antonio Miguel Quintales
Vidal Moreno Rodilla
Roberto Therón Sánchez
Ángel Francisco Zazo Rodríguez

Índice general

Representación Visual de los Sistemas de Software: Evolución y Colaboración	1
<i>Antonio González Torres, Francisco J. García Peñalvo y Roberto Therón Sánchez</i>	
Tools for Teaching Logic	25
<i>Juan Carlos Uclés Mármol y María Gracia Manzano Arjona</i>	
Organizaciones virtuales de agentes para la integración de personas discapacitadas	39
<i>Alberto López Barriuso y Juan Manuel Corchado Rodríguez</i>	
Plataforma multiagente adaptable al contexto sobre sistema <i>cloud</i>	57
<i>Víctor Parra Vidales, Juan Francisco de Paz Santana y Belén Pérez Lancho</i>	
Visualización de la evolución del software en sistemas de control de versiones	75
<i>Rodrigo Olmo Rodríguez y Roberto Therón Sánchez</i>	
Sistema Multiagente eficiente para la extracción de características y análisis de imágenes	83
<i>Isabel Martín Rodríguez, Juan Francisco De Paz Santana y Belén Pérez Lancho</i>	
Minería de datos aplicada al análisis de datos RNA-seq	99
<i>Óscar González Velasco y Luis Antonio Miguel Quintales</i>	
Sistema de predicción de edad en rostros	125
<i>Alfonso González Briones, Juan Manuel Corchado Rodríguez y Juan Francisco de Paz Santana</i>	
Sistema para la Distinción de Genero en Imágenes	137
<i>Francisco Javier Esteban Vicente y Juan Manuel Corchado Rodríguez</i>	
Arreglando y analizando textos	151
<i>Antonio Juan Sánchez, Emilio S. Corchado Rodríguez y Sara Rodríguez González</i>	
Automatic image analyzer to assess retinal vessel caliber (Altair)	161
<i>Pablo Chamoso Santos, Angélica González Arrieta y Sara Rodríguez González</i>	

Autores 183

Representación Visual de los Sistemas de Software: Evolución y Colaboración *

Antonio González Torres^{1,2}, Francisco J. García Peñalvo¹ y Roberto Therón Sánchez¹

¹ Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España

{agtorres, fgarcia, theron}@usal.es

² Escuela de Ingeniería, Universidad Latinoamericana de Ciencia y Tecnología. Urbanización Tournón, 10235-1000, San José, Costa Rica

agtorres@usal.es

Resumen El desarrollo y mantenimiento de software es un proceso dinámico y complejo que usualmente se extiende durante varios años, y que en la actualidad es común que se lleve a cabo en diferentes localidades geográficas, situadas en varios países y continentes. Esto supone para los equipos de trabajo un reto adicional (a los retos técnicos), por las distancias geográficas, horarias y culturales que existen. De forma que se acentúa la necesidad de contar con mecanismos para facilitar la colaboración entre los miembros de los equipos y conocer las actividades que se realizan en torno al sistema. En esta investigación se lleva a cabo un análisis detallado al respecto y se define un framework que tiene como objetivo describir y facilitar la discusión sobre los factores y aspectos que intervienen en la creación de la conciencia situacional de los equipos de trabajo y de un espacio de conocimiento situacional para facilitar la colaboración durante los procesos de desarrollo, mantenimiento y evolución de software. Con base en la definición de dicho framework y las discusiones que se llevan a cabo sobre el uso de la visualización de software como medio de apoyo, se presenta y explica en detalle una herramienta de visualización que fue diseñada e implementada con el fin de contribuir con los procesos mencionados.

Keywords: Visualización de software, visualización de la evolución de software, arquitectura de sistemas, conciencia situacional, colaboración

1. Introducción

En la actualidad es común que un gran número de empresas lleven a cabo los procesos de Desarrollo, Mantenimiento y Evolución de Software (DMES) utilizando esquemas de Desarrollo Global de Software (DGS). Esto implica que los miembros de los equipos de trabajo se encuentran distribuidos en diferentes

* Este trabajo fue apoyado por el Ministerio de Ciencia, Tecnología y Telecomunicaciones (MICITT) de Costa Rica.

áreas geográficas [1, 2, 3, 4] y que durante la planificación de los proyectos es necesario considerar tanto las distancias geográficas como las temporales y socio-culturales.

En general dichas distancias requiere que los equipos de trabajo se enfrenten a diversos retos relacionados con la comunicación, coordinación y control [5, 6, 7, 8] que se requiere para contar con conocimiento oportuno sobre las actividades que se llevan a cabo [9, 10, 11, 12], así como para comprender el estado de los proyectos de software y hacer posible que su desarrollo y mantenimiento se realice de forma adecuada y eficiente.

Tomando en consideración lo anterior, los equipos de trabajo requieren contar con mecanismos efectivos para la comunicación, coordinación y control entre sus miembros [5, 13, 9]; siendo algunos de esos mecanismos los siguientes:

- Documentación actualizada de los sistemas (requerimientos, especificación, diseño y manuales).
- Metodologías de desarrollo de software apropiadas.
- Email, teléfono, mensajería instantánea y sistemas de video-conferencias.
- Tecnologías y herramientas para compartir y obtener conocimiento sobre las actividades que llevan a cabo los miembros del equipo de trabajo.
- Estrategias de formación de equipos.

En este contexto otro factor que es necesario tener en cuenta es el tipo de organización que los equipos de trabajo utilizan, de forma independiente a la estructura organizacional usada por la compañía a la que pertenecen [14], siendo los siguientes los tipos de organización más comunes:

- Equipos virtuales [15, 5].
- Equipos coherentes y con ubicación compartida de programadores [16, 17].
- Equipos débilmente acoplados [6, 18].
- Desarrollo global de software con un enfoque disperso [19].

A todo lo anterior se deben agregar las características dinámicas de los procesos de DMES. Cada programador trabaja en un ciclo que requiere la comprensión del estado actual del sistema, luego, con base en esa comprensión realiza la programación de código fuente y evalúa los efectos de los cambios que ha efectuado, así como los cambios que puedan afectarles en su trabajo y que hayan sido realizados por otros miembros de su equipo de trabajo; para luego volver al punto de partida: hacer más cambios al sistema hasta satisfacer los requerimientos de desarrollo, mejora o mantenimiento. El objetivo es que el sistema evolucione, siga siendo susceptible de mantenimiento en el largo plazo y satisfaga las necesidades de los usuarios [20].

Con base en esto se puede notar que los programadores requieren de conocimiento actualizado sobre las actividades y cambios que sufre el sistema con el fin de comprenderlo y que sea posible efectuar su desarrollo y mantenimiento. En general, los programadores hacen uso de herramientas diversas, como los Entornos de Desarrollo Integrados (IDEs), compiladores, depuradores, bug trackers

y herramientas de Administración de la Configuración de Software (SCM) para llevar a cabo sus tareas.

Los IDEs son un elemento valioso para capturar información sobre las actividades que realizan los programadores, al igual que las herramientas SCM. Es conveniente considerar que estas últimas herramientas son usadas de forma amplia en los procesos de DMES para facilitar la interacción y colaboración entre los desarrolladores, lo que tiene como consecuencia el registro de detalles en los repositorios de software sobre las actividades que son llevadas a cabo por los programadores. En este escenario el análisis automático de la información es un elemento clave que hace posible la extracción de hechos relevantes sobre las actividades de los programadores, y los cambios recientes e históricos que han efectuado al sistema. Sin embargo, la comprensión del resultado de este tipo de análisis usualmente es complejo por las grandes cantidades de información que produce y por la naturaleza de los sistemas de software, los cuales están conformados por un gran número de elementos (e.g. paquetes, elementos de software, métodos y atributos) que se encuentran relacionados entre sí (e.g. herencia, implementación de interfaces y dependencias) y que cambian con cada versión del sistema.

La visualización de la información ha demostrado que mediante su uso es posible representar grandes cantidades de información, y en especial, la visualización de software ha mostrado su utilidad en facilitar la comprensión de los sistemas [21], los cambios que se les practican y su evolución. En síntesis, el uso de la visualización de software facilita la comprensión del estado actual del sistema, usando información reciente y de su evolución, la cual puede comprender un periodo de tiempo reciente o la evolución completa del sistema. Por lo que mediante su uso es posible crear un espacio de trabajo que ponga a disposición de los equipos de desarrollo la información sobre el estado de las cosas ³ en torno al sistema en que trabajan y durante todos los procesos que comprenden los procesos DMES .

En la creación de ese espacio de trabajo se debe tener en cuenta que la arquitectura es el elemento central de los sistemas y que la mayoría de tareas que realizan los equipos de trabajo giran en torno a esta, por lo que su comprensión, y la de los elementos que la componen, es determinante durante los procesos de DMES. Por lo tanto, la visualización de la evolución de la arquitectura de los sistemas de software, así como de las relaciones de herencia entre clases, implementación de interfaces, métricas, colaboración entre programadores y relaciones socio-técnicas son características deseables de una herramienta que busque apoyar la conciencia situacional y la colaboración en un entorno de DGS.

Este trabajo de investigación se enfoca en la visualización de sistemas de software y su evolución, y tiene como objetivo facilitar la colaboración entre los miembros del equipo de desarrollo brindando información sobre las actividades que se llevan a cabo en torno a la arquitectura de los sistemas y su evolución, teniendo en cuenta:

³ En inglés se utiliza la palabra *awareness* para referirse al conocimiento de lo que sucede en un sistema o entorno.

1. El escenario actual del desarrollo de software en diferentes localidades geográficas (DGS).
2. Las necesidades de información de los Administrador de Proyectos (APs) y programadores sobre las actividades que realizan los miembros del equipo de trabajo (para controlar y monitorear las actividades) y la evolución de la calidad del sistema y sus elementos (medida con el uso de métricas).
3. La ausencia de un framework para describir la interacción entre los aspectos que intervienen durante los procesos de DMES y su relación con la necesidad de diseñar espacios de conocimiento compartido para facilitar la toma de decisiones y colaboración entre los miembros del equipo de trabajo en entornos DGS.

El resto de este documento está organizado de la siguiente forma: la sección 2 hace un análisis sobre el trabajo en equipo, la arquitectura de los sistemas de software y el uso de la visualización para facilitar la conciencia situacional y la colaboración; la sección 3 discute y explica algunas consideraciones para el diseño de espacios para facilitar la creación de la conciencia situacional y la colaboración entre los miembros de los equipos de trabajo; la sección 4 presenta y explica la propuesta de un framework que muestra las relaciones entre los elementos que intervienen en la creación de la conciencia situacional y la colaboración; la sección 5 presenta GridMaster, una herramienta implementada en Java como un plugin de Eclipse para la visualización de la estructura de los sistemas y la colaboración entre miembros del equipo de trabajo, y finalmente la sección 6 presenta las conclusiones de este trabajo.

2. Antecedentes

Este apartado está organizado en tres secciones y tiene como objetivo realizar el análisis sobre el trabajo en equipo (ver sección 2.1), la arquitectura de los sistemas de software (ver sección 2.2) y el uso de la visualización para facilitar la conciencia situacional y la colaboración (ver sección 2.3).

2.1. Trabajo en equipo

Las organizaciones han adoptado el trabajo en equipo para el desarrollo de software, con preferencia sobre el trabajo individual, porque consideran que el funcionamiento efectivo de los grupos de trabajo puede proporcionar buenos resultados [22] por la diversidad de sus miembros, en términos de experiencia y especialización, aunque conviene recordar las diferencias de nacionalidad, cultura y localización geográfica de estos cuando se trabaja con el modelo de DGS (estas diferencias pueden constituir ventajas y desventajas) [23, 24].

Para que un equipo de trabajo funcione de forma adecuada quienes lo conforman deben ser capaces de trabajar juntos de forma efectiva [23]: es necesario que los miembros del equipo se comuniquen, coordinen y trabajen juntos de forma eficiente, particularmente cuando se usa un modelo DGS. Por lo que es importante que los miembros de los equipos, y los equipos como tales, cuenten con la

información que requieren de acuerdo con las tareas que llevan a cabo [24]. La importancia de lo anterior radica en que al final del día los módulos y diferentes elementos del sistema deben integrarse y funcionar en conjunto, y por lo general la forma como se realiza la asignación de tareas a los grupos de trabajo e individuos es tomando como base la arquitectura del sistema y los módulos que la componen [13].

Comunicación, coordinación, control y cognición Los factores de comunicación, coordinación y control se encuentran estrechamente relacionados entre sí, pero a estos es importante agregar el factor “cognición” de acuerdo con el trabajo efectuado por Comfort [25]. La gestión de emergencias [25] tiene varias similitudes con la naturaleza cambiante y dinámica (a veces poco predecible) del desarrollo de sistemas de software: el desarrollo de software requiere de la rápida adaptación de los actores de acuerdo a los eventos que se presentan y los nuevos escenarios que se originan, particularmente cuando su desarrollo se lleva a cabo de forma distribuida.

Cognición: Usando con algunas modificaciones las definiciones proporcionadas por Comfort [25], la cognición se define como un proceso que depende de un modelo mental claro sobre como debe funcionar el sistema bajo observación, y por lo tanto es el elemento que activa los procesos de comunicación, coordinación y control cuando se detectan discrepancias entre lo que es visto por los individuos como un desempeño normal y los cambios de estado de los indicadores claves que alertan sobre desviaciones potenciales del proceso.

Comunicación: El fin de la comunicación es “comunicar” o hacer partícipe a otros de algo mediante un lenguaje común. El diccionario The Merriam-Webster define comunicación como “un proceso mediante el cual la información es intercambiada entre individuos a través de un sistema común de símbolos, señales o comportamiento” [26].

Coordinación: La coordinación puede ser implícita o explícita. La coordinación implícita depende del conocimiento del equipo de trabajo y su capacidad para tomar decisiones en situaciones críticas con un nivel reducido de comunicación, por lo que los miembros del equipo ajustan sus comportamientos de forma dinámica para anticiparse a las acciones y atender de forma proactiva las tareas que así lo requieren [27, 22]. La coordinación implícita puede proporcionar ventajas cuando la carga de trabajo es muy alta, debido a que se requiere menos comunicación, y está asociada a equipos con alto desempeño en los cuales los miembros entienden de forma clara las necesidades y responsabilidades de sus tareas [22]. En tanto la coordinación explícita es el proceso de organizar las cosas, personas o grupos para que trabajen juntos de forma apropiada [26]. Para efectuar las labores de coordinación es necesario contar y utilizar mecanismos que permitan los procesos de intercambio efectivo de información y entendimiento para alinear las prioridades y acciones de los diferentes actores para alcanzar una meta compartida [25, 28, 22]. Las ventajas y desventajas de usar uno u otro tipo de coordinación depende de las circunstancias y las tareas que se llevan a cabo [22]. Cabe agregar

que los mecanismos de coordinación adquieren una importancia particular cuando se utiliza un enfoque DGS por las distancias (geográficas, culturales y horarias) que existen entre los diferentes sitios involucrados en el desarrollo del proyecto [10].

Control: Tomando en consideración los diversos factores que causan incertidumbre durante el proceso de desarrollo de software, el control se puede definir como la capacidad de mantener las acciones enfocadas en alcanzar las metas y objetivos fijados [25, 26] para producir un producto de software correcto, a tiempo y dentro del presupuesto [29]. Lo anterior, de acuerdo a los requerimientos, especificaciones, plazos, costos, estándares, políticas, niveles de calidad y otros factores inherentes al proceso de software que al ser atendidos de forma oportuna facilitan lograr el éxito del proyecto. Además, conviene considerar la dificultad para realizar el control cuando se utiliza el modelo DGS [30] por los factores discutidos en los apartados anteriores.

Conciencia situacional del equipo El trabajo en equipo permite que los integrantes de un equipos se familiaricen con los demás miembros y conozcan sobre los conocimientos, habilidades, experiencia, contexto, personalidades y hábitos de cada uno. Ese conocimiento mutuo varía con el tiempo y se incrementa conforme este pasa, lo que hace posible una mejor planificación del trabajo [22].

Endsley define la conciencia situacional como la percepción de los elementos en el ambiente en un periodo de tiempo y espacio determinado, así como su comprensión y la proyección de esos elementos en un futuro cercano [31, 32, 33]. En esta investigación conciencia situacional es definida como el grado de conocimiento sobre el estado de las tareas, actividades, cambios y resolución de problemas relacionados con el proceso de desarrollo, mantenimiento y evolución de un sistema de software. En esta investigación se considera que la conciencia situacional tiene importancia tanto a nivel individual como del equipo de trabajo en los procesos de software.

La forma como los individuos procesan la información y construyen su modelo mental de conciencia situacional depende de sus objetivos, habilidades, experiencia, capacitación y rol desempeñado en los procesos de DMES. Por lo que los administradores de proyectos están interesados en los aspectos de más alto nivel [34] y los programadores en los detalles más específicos.

La experiencia de los individuos y el equipo de trabajo en sistemas similares son de gran valor en la construcción de la conciencia situacional, principalmente en cuanto al conocimiento de las habilidades de los demás miembros del equipo y la asignación de tareas [23], pero también en torno a la mecánica de funcionamiento del equipo. Sin embargo, cada proyecto de software es único por las diferencias de los problemas que buscan resolver, los enfoques que se usan para resolverlos y la interdependencia de los elementos internos del sistema. Por lo que aunque un sistema de software tenga elementos comunes en relación con otro, los problemas a los que se enfrentan los individuos y el equipo de trabajo, así como las variables a tomar en cuenta, son diferentes. De forma que la construcción de

la conciencia situacional requiere de tiempo, por ser necesaria la acumulación de experiencia con el proyecto en que trabajan el equipo y quienes lo conforman.

Conciencia situacional distribuida Otro enfoque más reciente, que es complementario con el mencionado en la sección anterior, es conocido como Espacio de Trabajo de Conciencia Situacional Distribuida (ETCSD), el cual tiene una orientación basada en sistemas en la cual los individuos y los elementos tecnológicos son considerados como agentes, que interactúan, tienen diferentes propósitos (por las tareas y actividades que llevan a cabo) y su propia Conciencia Situacional (CS) sobre las tareas que realizan y el proyecto en general. La idea detrás es que los miembros de un equipo de trabajo no necesitan conocer todos los detalles sobre el proyecto en que trabajan, sino solo aquellos detalles que les permita realizar sus tareas. Pero implica que deben tener conciencia sobre lo que sucede con el proyecto de forma general y saber la información que los demás requieren y necesitan conocer sobre las tareas que tiene bajo su responsabilidad [35].

Cabe mencionar que el acceso a la misma información no produce la misma conciencia situacional en los miembros de un equipo, por las metas, tareas, roles y experiencia de cada uno, que hace que la usen de forma diferente a los demás. Por lo que el conocimiento se encuentra distribuido en el entorno y la CS de un agente puede ser diferente pero compatible con la CS de otros agentes. La realización de determinadas tareas que se encuentran relacionadas entre sí requieren la colaboración entre los agentes para que las lleven a cabo de forma coordinada o conjunta, por lo que la compatibilidad de la CS de los agentes resulta de gran utilidad.

2.2. Arquitectura de los sistemas de software

La arquitectura de un sistema de software tiene como fin mostrar una vista abstracta del sistema y puede ser diseñada usando capas con diferentes niveles de detalle. El nivel de detalle de las capas depende del propósito de la arquitectura, los requerimientos del sistema y el entorno en que funcionará el sistema. Una arquitectura de alto nivel es útil para comunicar una visión general a los gerentes, administradores de proyectos y usuarios, mientras que una arquitectura de bajo nivel sirve para orientar el diseño detallado del sistema y apoyar a los programadores cuando aún no se han familiarizado con el sistema [36], al mostrar, por ejemplo, información sobre las relaciones entre los elementos de software y las estructuras de datos a utilizar. Tomando en cuenta lo anterior, es posible hacer varios diseños de la arquitectura para el sistema, según su propósito.

Es importante resaltar que a partir de la arquitectura es posible comprender como está organizado un sistema de software en términos de modularidad, componentes, relaciones entre elementos, estructuras de datos, acceso a datos y la distribución física de los componentes del sistema en servidores y equipos de usuarios.

En la actualidad la mayoría de sistemas de software se desarrollan usando lenguajes orientados a objetos y utilizan una estructura jerárquica (packages,

classes, methods and attributes). Como consecuencia, el trabajo realizado en esta investigación está basado en el análisis de sistemas de software desarrollados con el paradigma de programación orientada a objetos.

Otra consideración con respecto a la arquitectura de los sistemas es que el proceso de aseguramiento de la calidad del software mide mediante el uso de métricas los elementos que la componen (e.g. tamaño, complejidad, dependencias y relaciones), y el objetivo particular de las métricas de evolución es permitir la comparación y evolución de la calidad de varias revisiones o periodos de tiempo.

En línea con esto, Khan *et al.* [37] realizaron una investigación sobre el uso de la visualización para representar los elementos relacionados con la arquitectura de los sistemas. De acuerdo con esa investigación los elementos más comunes que son representados por las visualizaciones de software, tomando en cuenta una o más revisiones del sistema, son los siguientes:

- Acoplamiento y acoplamiento lógico.
- Dependencias.
- Métricas y métricas de evolución.
- Relaciones entre elementos de software (e.g. herencia e implementación de interfaces).
- Estructura y cambios en la estructura.
- Vocabulario.

2.3. Visualización para facilitar la conciencia situacional y la colaboración

En la mayoría de los casos, el objetivo de las herramientas que apoyan la construcción de espacios comunes de conocimiento es mejorar la cooperación entre los miembros del equipo. Dicha cooperación usualmente es necesaria porque los colaboradores trabajan en un proyecto común y el trabajo de cada uno tiene relación con el trabajo de otros colaboradores. Pero es conveniente considerar que la cooperación entre individuos también está presente cuando varias personas aportan a la solución de determinados problemas, trabajando de forma independiente y sin tener como fin la consecución de objetivos o metas comunes. Un ejemplo notable son los sitios web utilizados por los programadores para solicitar la colaboración de otros programadores en la solución de problemas particulares [38]. Los programadores que participan en este tipo de sitios comparten la pertenencia a una comunidad de individuos que trabajan en temas sobre los cuales tienen conocimiento, pero que necesariamente no forman parte de su trabajo diario.

Assogba y Donah [38] dieron el nombre de “loosely bound cooperation” a este tipo de cooperación y lo definieron como “una forma de cooperación , a veces indirecta, entre miembros de una comunidad que les da libertad para perseguir sus metas individuales a la vez que les permite ayudarse entre sí”. Como un resumen de todo lo anterior y de las características que los mismos investigadores resaltan, los siguientes son los puntos centrales de este tipo de cooperación:

- Los individuos no adquieren ningún tipo de obligación de ayudar a otros.

- Cada participante tiene sus propias metas y de forma mayoritaria no tienen metas compartidas.
- La cooperación puede ser desde casual hasta continua y comprometida e involucra la solución o desarrollo de un determinado elemento de software.
- Los participantes de este modelo de cooperación forman parte de una comunidad de individuos que practican de forma activa su profesión.

Durante el trabajo de investigación realizado por Assogba y Donah desarrollaron una herramienta en Processing a la cual denominaron con el nombre de Share [38]. El objetivo de esa herramienta es apoyar el intercambio de código fuente entre los miembros de una comunidad de programadores. Para su implementación utilizaron una arquitectura cliente/servidor, la cual del lado servidor proporciona la autenticación y almacenamiento de datos, mientras que del lado cliente es una aplicación de escritorio en donde el usuario realiza las tareas de programación. A cada programador que utiliza esta herramienta se le asigna un color que lo representa en todos los proyectos que participa. El lado cliente de Share proporciona un navegador de archivos, un editor de programas, un gestor de referencias, un buscador, la visualización de la red de relaciones (navegador de relaciones) y mecanismos para la sincronización con el servidor.

La visualización de la colaboración entre programadores permite que un programador obtenga información sobre quienes le pueden prestar colaboración a partir de los elementos que han cambiado y las relaciones que existen entre ellos. Este tipo de visualización ofrece información a los administradores de proyectos para tomar decisiones sobre cuál programador puede sustituir a otro programador en caso de baja por enfermedad, accidente, renuncia o despido, pero además también les puede ayudar a conformar equipos de trabajo de acuerdo a las relaciones de colaboración previa entre los individuos [39].

Jermakovics *et al.* [39] construyen la red de colaboración que tiene lugar entre los desarrolladores usando información que extraen del repositorio de software sobre los cambios que se realizan a los elementos de software. Dicha red de colaboración emerge como producto del análisis de las similitudes entre los programadores con base en los elementos de software que han cambiado en común.

Una propuesta complementaria a las anteriores es la que realizan Heller *et al.* [40] sobre la visualización de la colaboración entre programadores, pero tomando en consideración su ubicación geográfica. En esta visualización se representan datos que son obtenidos de GitHub por medio de un grafo que se dibuja sobre un mapa para mostrar las relaciones entre los programadores, lo que además permite mostrar la densidad de programadores por países o regiones.

Elaborar estrategias para incrementar el nivel de conocimiento sobre las actividades que realizan los programadores puede contribuir con la coordinación en entornos de desarrollo distribuido. Teniendo esto en consideración, conviene recordar que las herramientas SCM han sido ampliamente difundidas y utilizadas para contribuir en la coordinación del desarrollo paralelo de software y que en los entornos distribuidos han sido de gran utilidad. Además, por la riqueza y gran cantidad de información que gestionan, un gran número de herramientas de visualización hacen uso de esa información. Pero las herramientas SCM tienen

como desventaja que los programadores se dan cuenta de las modificaciones de los demás programadores hasta que han sido enviadas al repositorio de software mediante una operación “check-in”, y no en el momento en que se realizan los cambios [41].

Considerando lo anterior, Lanza *et al.* proponen una arquitectura que utiliza un plugin de Eclipse para grabar y transmitir los cambios de código que se llevan a cabo en el entorno de un programador, a todos los programadores. Para apoyar la comprensión y reacción frente a los cambios desarrollaron otro plugin que hace uso de tres visualizaciones simples que se actualizan en tiempo real conforme llega la información de los cambios (para más detalles sobre las visualizaciones puede referirse a [41]).

Mientras que *Ownership Map* [42, 43] es una visualización que brinda información a los administradores de proyectos sobre la colaboración que ha tenido lugar durante el desarrollo del sistema. El objetivo de dicha representación es apoyar la toma de decisiones y ofrece detalles sobre:

- El número de programadores que han participado en el desarrollo del sistema.
- Las modificaciones o partes del sistema que han sido desarrolladas por cada programador.
- El comportamiento de los programadores durante el desarrollo y mantenimiento del sistema.

Los datos usados por *Ownership Map* se extraen de los logs de Concurrent Versioning System (CVS) y contemplan detalles sobre la asociación de los cambios, a nivel de líneas de código, y los programadores. Con base en esos datos se determina cuál programador es el propietario de un elemento de software durante un intervalo de tiempo o durante la evolución completa de ese elemento.

3. Consideraciones para diseñar espacios para la conciencia situacional

Tomando en cuenta los conceptos y definiciones enunciadas, realizar el diseño de un espacio para facilitar la construcción de la conciencia situacional del equipo de trabajo requiere considerar que dicho espacio, además de proporcionar detalles que permitan comprender la situación actual, debe tomar en cuenta los hechos pasados para obtener información sobre como se llegó a un determinado estado, pero además debe apoyar la realización de proyecciones futuras a partir del análisis de esa información.

Para el diseño de un Espacio de Conciencia Situacional (ECS) es necesario identificar los elementos que el individuo o equipo deben conocer, de acuerdo a los objetivos y metas del proyecto, y las tareas o actividades que les han sido encomendadas. Entonces, para apoyar el proceso de mantenimiento de software, por ejemplo, resulta de utilidad que dicho espacio brinde información que ayude a comprender los cambios que se realizan en la arquitectura del sistema, lo cual puede contribuir a mejorar el desempeño de los individuos y el equipo en general [44, 22].

Para realizar el diseño de un ECS es importante tener en consideración los puntos que se presentan a continuación:

- Es común que los proyectos de software se lleven a cabo utilizando un modelo DGS.
- La meta de un ECS es apoyar la construcción de un Conciencia Situacional del Equipo de Trabajo (CSET) con el fin de facilitar la colaboración entre los miembros del equipo cuando llevan a cabo las tareas y actividades bajo su responsabilidad.
- DMES es un proceso dinámico cuyo estado cambia de forma constante, lo que dificulta mantener actualizada la información sobre las actividades, tareas y patrones de progreso que hayan sido determinadas como de interés durante el diseño del ECS.
- En entornos complejos y cambiantes, como el de DMES, las decisiones se deben tomar en poco tiempo, por lo que se requiere de información actualizada.
- Contar con información actualizada sobre el estado de las cosas es importante incluso cuando los miembros del equipo realizan tareas triviales.
- Un ECS debe brindar información sobre el estado presente de las cosas, pero es necesario considerar aquellos elementos que puedan revelar cambios futuros del estado del proceso con el fin es ayudar a decidir mejor el curso de las acciones a realizar.
- Contar con información errónea sobre el estado de las cosas puede llevar a tomar decisiones equivocadas.
- Los miembros del equipo requieren conocer lo que ocurre en relación con sus tareas y el proyecto en general, pero además deben ser capaces de interpretar correctamente las diferentes situaciones de acuerdo a las metas del proyecto para que con base en estas realicen la toma de decisiones y lleven a cabo las acciones pertinentes de forma oportuna (entender la situación →tomar decisiones →llevar a cabo acciones).
- La falta de capacitación de los miembros del equipo o que no cuenten con las habilidades requeridas puede conducir a una mala interpretación de CS [44, 31], lo cual puede tener como consecuencia la toma incorrecta de decisiones.
- La conformación de equipos de trabajo con estructuras cohesivas en términos de conocimiento, habilidades y control de actividades es un objetivo difícil de alcanzar.

El uso de la visualización de software surge, entonces, como una alternativa viable para construir un ECS por sus capacidades para transmitir información y proporcionar mecanismos de interacción con los usuarios. Las herramientas de visualización pueden hacer posible la interacción entre usuarios mediante métodos de anotación o simplemente al proporcionar información sobre lo que sucede a los miembros de equipo de trabajo para que inicien las tareas de comunicación, coordinación y control a partir de los procesos de cognición que se activan al adquirir conciencia sobre la situación.

4. Framework: conciencia situacional y colaboración

Los procesos de DMES son complejos, dinámicos, impredecibles y es común que se lleven a cabo en diferentes localidades geográficas y durante varios años. Lo que requiere que los miembros de los equipos de trabajo establezcan relaciones de colaboración de largo plazo, o puntuales, para realizar tareas y solucionar problemas concretos. En cualquiera de los casos cuando una tarea ha sido terminada o un problema ha sido resuelto, los individuos pasan a trabajar en lo siguiente que tienen en la agenda de acuerdo a la planificación del proyecto o en la lista de problemas pendientes de resolución. Esto conlleva que la colaboración no se lleva a cabo siempre entre los mismos miembros del equipo, sino en función de especialización y las tareas que acometen en un momento determinado.

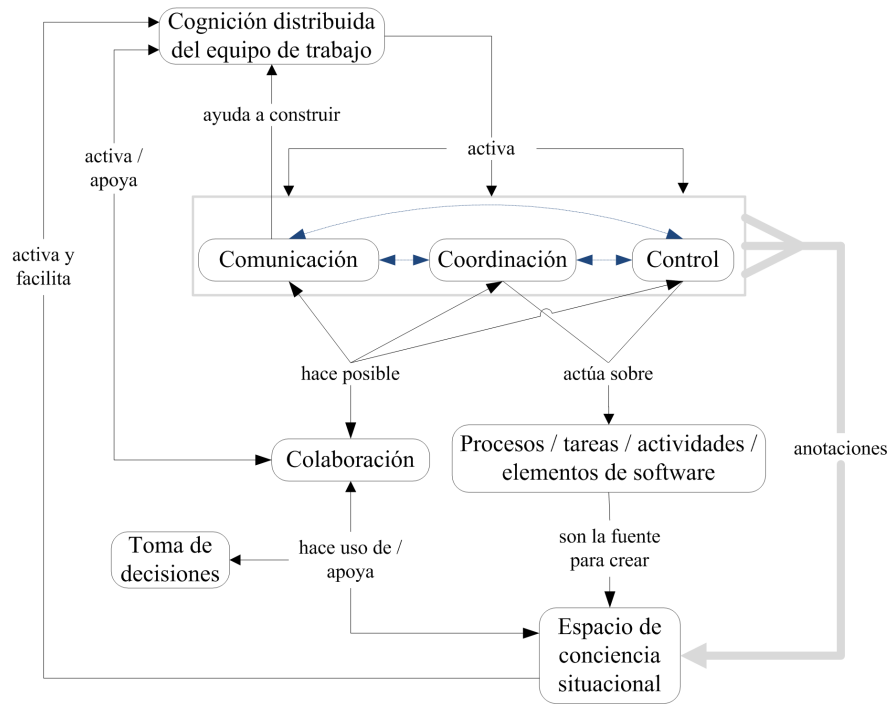


Figura 1. Framework sobre el trabajo colaborativo en los procesos de DMES.

La colaboración entre los miembros de un equipo de trabajo puede iniciar por diferentes circunstancias o situaciones, lo que implica que no existe una relación de orden entre los aspectos relacionados con ese proceso (ver la sección 2.1). De acuerdo con esto, el framework que se ilustra en la Figura 1 tiene como fin orientar sobre una posible configuración de las relaciones entre esos aspectos, así como definir el papel que pueden jugar los ECSs en los procesos de colaboración.

En la Figura 1, las palabras de enlace compuestas (e.g. activa / apoya) sobre las líneas de enlace con flechas bidireccionales se leen siguiendo un orden descendente / ascendente. En el caso del orden descendente se inicia por el concepto en el extremo superior de la línea y se termina con el concepto en el extremo inferior (e.g. Cognición distribuida del equipo de trabajo – activa – Colaboración), mientras que en el caso del orden ascendente, se inicia a la inversa (e.g. Colaboración – apoya – Cognición distribuida del equipo de trabajo).

El framework de la Figura 1 ha sido definido tomando en consideración un enfoque de cognición distribuida en el que los miembros de los equipos de trabajo hacen uso de su cognición individual, de acuerdo a su especialización y experiencia, y la complementan con la cognición de los demás individuos para actuar en la realización de tareas y la resolución de problemas. Con base en esto, el proceso de colaboración puede ser activado por uno o varios miembros del equipo de trabajo haciendo uso de sus capacidades cognitivas cuando requiere llevar a cabo una tarea o detecta una situación ante la cual debe actuar.

En este contexto, para que la colaboración sea posible se puede requerir de las actividades de comunicación, coordinación o control entre los miembros del equipo de trabajo. Es importante tener presente que la colaboración entre individuos requiere de información para la distribución de tareas y para determinar las acciones que se deben seguir de acuerdo al estado del proyecto. Por lo que las facilidades e información que proporcionan los ECSs sobre la evolución y el estado de los procesos, tareas, actividades y cambios que se han realizado a los elementos de software del proyecto son de gran utilidad. La relación que existe entre el proceso de colaboración y las actividades de comunicación, coordinación y control es recíproca, como muestra la Figura 1.

De forma análoga a la activación de la colaboración, uno o varios miembros del equipo de trabajo pueden activar las actividades de comunicación, coordinación o control, para que a su vez se active el proceso de colaboración entre las personas adecuadas, y en caso necesario se haga uso de las ventajas de la utilización de los ECSs.

En este punto es relevante mencionar que el fin de los ECSs es ofrecer información en torno a los procesos, tareas, actividades y artefactos de los sistemas de software, pero además brindar la posibilidad de interacción a los usuarios para que puedan efectuar anotaciones en las visualizaciones durante las actividades de comunicación, coordinación y control (ver Figura 1). El propósito de los ECSs es apoyar los procesos de colaboración y toma de decisiones, por lo que su diseño puede tomar en consideración varios de los siguientes objetivos:

- Proporcionar información sobre el estado del sistema para facilitar la realización de una tarea o la resolución de un problema de mantenimiento.
- Facilitar la construcción de la cognición individual y grupal en torno a los procesos de DMES de un proyecto determinado al brindar información de seguimiento del estado del sistema.
- Mostrar advertencias o presentar patrones que dejen al descubierto un comportamiento no esperado de los cambios o actividades que se han llevado

a cabo en el sistema para activar los mecanismos de colaboración entre los miembros del equipo de trabajo.

- Apoyar a los administradores de proyectos en los procesos de toma de decisiones, que pueden estar relacionadas con la asignación de tareas a los miembros del equipo, las actividades y tareas en proceso o con aspectos muy concretos y técnicos de los elementos de software del sistema.
- Asistir a los programadores en su trabajo individual y el cumplimiento de tareas, metas y objetivos que le han sido asignados al ofrecer detalles sobre el progreso de las actividades que están realizando los demás miembros del equipo y el estado general del proyecto.

Cabe resaltar que las actividades de coordinación y control tienen efectos directos sobre los procesos, actividades y elementos de software del sistema, los cuales son la fuente que se utiliza para alimentar las visualizaciones de los ECSs.

5. GridMaster: Visualización de la estructura de los sistemas y colaboración

GridMaster es una visualización para la evolución de proyectos de software que fue programada en Java como un plugin de Eclipse. Los objetivos de GridMaster son apoyar a los administradores de proyectos y programadores y proporcionar información para facilitar la colaboración entre los miembros de los equipos de trabajo. El desarrollo de esta herramienta se llevó a cabo tomando en cuenta los requerimientos [45, 46] que se presentan a continuación:

Audiencia: Apoyar tanto a los administradores de proyectos como a los programadores mediante información pertinente de acuerdo a las tareas que desempeñan.

Tareas: Dar soporte a las siguientes tareas para apoyar a administradores de proyectos y programadores:

Toma de decisiones: Facilitar la toma de decisiones sobre la asignación de tareas a los programadores.

Comprensión de relaciones: Proporcionar información que permita comprender las relaciones entre los programadores y los elementos de software del sistema, así como la relación entre los mismos programadores.

Entender la estructura: Apoyar la comprensión de los cambios y la evolución de la estructura de los sistemas de software.

Comprensión de dependencias: Brindar información sobre las relaciones de herencia e implementación de interfaces y los cambios de esas relaciones en el tiempo.

Apoyo a la calidad: Ofrecer detalles sobre las variaciones de la calidad de los elementos de software mediante el uso de métricas de evolución.

Datos: Realizar la recuperación y análisis automático de los logs y el código fuente almacenado por los repositorios de software gestionados por las herramientas SCM.

Representación: Hacer uso de una representación sencilla y fácil de usar que transmita la información de forma rápida y con poca carga cognitiva, y que además soporte técnicas de interacción para la navegación y exploración de los datos. De forma adicional la visualización debe ofrecer:

Escalabilidad: La visualización debe ser capaz de representar la estructura de grandes sistemas de software y de acomodar el crecimiento de esa estructura conforme el sistema evoluciona.

Estructura estable: La visualización debe conservar la representación de los elementos de la estructura del sistema en la misma posición desde las primeras revisiones hasta las últimas para brindar al usuario una visualización uniforme durante la evolución del proyecto.

5.1. Descripción de la visualización

Con el uso de estructuras matriciales (ampliamente conocidas por los ingenieros de software) es posible diseñar representaciones visuales sencillas, escalables, poderosas e intuitivas que permiten correlacionar un gran número de elementos de datos y facilitan el descubrimiento de conocimiento. Las visualizaciones que utilizan este tipo de estructuras utilizan las filas y las columnas para representar datos que son correlacionados mediante la colocación de elementos gráficos en las celdas que se forman con su intersección.

GridMaster utiliza una estructura matricial cuyas celdas correlacionan la estructura con la evolución del sistema (ver Figura 2). En esta visualización la primera columna de la matriz es usada para representar la estructura del sistema, en donde los paquetes y elementos de software son distribuidos entre las filas de la matriz. Mientras que la primera fila de la estructura matricial es usada para representar la línea de tiempo de la evolución del sistema, y las unidades de tiempo que la conforman son distribuidas entre las columnas de la representación.

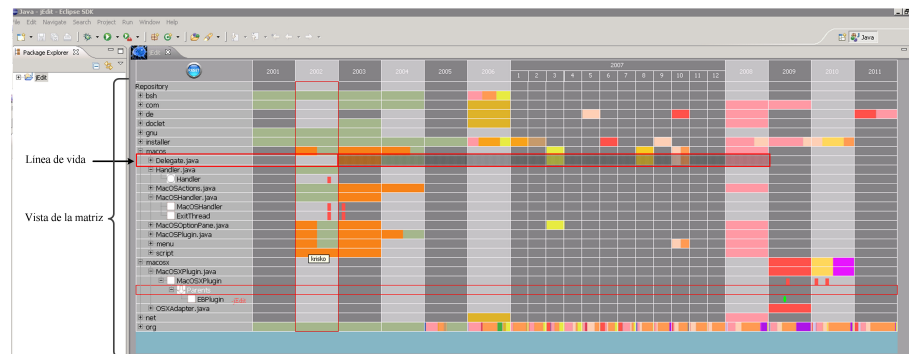


Figura 2. Vista principal de GridMaster.

Las celdas de GridMaster son usadas para correlacionar en el tiempo la estructura de un sistema con las métricas asociadas a sus elementos, las contribuciones de los programadores, la creación de los elementos de software, las relaciones socio-técnicas, y las creación o eliminación de las relaciones de herencia e implementación de interfaces.

Las técnicas de interacción utilizadas por esta visualización incluyen el uso del zoom, la distorsión Fisheye, la utilización de filtros para quitar nodos de la estructura del sistema y la posibilidad de seleccionar entre la representación absoluta o relativa de los valores de las métricas y contribuciones efectuadas por los programadores.

5.2. Estructura del proyecto

La representación visual de la estructura del proyecto se compone de los paquetes y elementos de software que han sido agregados durante su evolución. Si un paquete o elemento de software ha sido eliminado o movido de posición en la estructura del sistema este sigue siendo representado visualmente en su posición original, pero también es representado en su nueva posición. La correlación que se hace entre estos elementos y la línea de tiempo permite observar sus actividades en el tiempo, así si el elemento no existe en una determinada posición no se observará más actividad asociada a él. Esta característica construye una representación visual estable de la estructura del sistema, de forma que el usuario cuenta con una misma representación para toda la evolución del sistema. La Figura 3(a) muestra del lado izquierdo la estructura actual del sistema jEdit en el entorno de Eclipse, mientras que del lado derecho despliega la representación visual de la estructura del sistema incluyendo los paquetes y elementos de software que en la actualidad no forman parte del sistema.

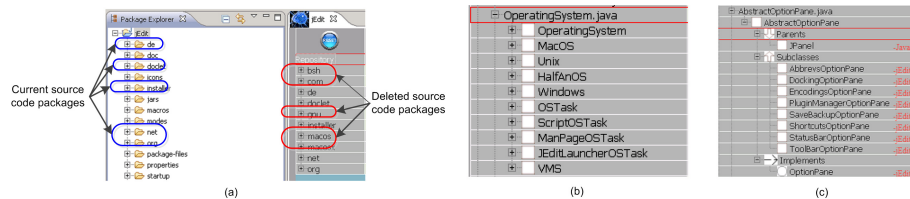


Figura 3. Estructura del sistema y contenido de los archivos. (a) Representación de la estructura del sistema. (b) Elementos de software que contiene un archivo. (c) Relaciones de herencia e implementación de interfaces.

5.3. Líneas de vida, colaboración y relaciones socio-técnicas

GridMaster representa las líneas de vida de los paquetes y elementos de software haciendo una correlación entre la estructura del sistema y la línea de tiempo, para lo cual rellena las celdas de la matriz en donde existe actividad de

forma proporcional al volumen de actividades. El relleno que es colocado en las celdas de la representación usa colores para indicar cuales programadores han llevado a cabo las actividades (las cuales son operaciones de “check-in” sobre los archivos usando una herramienta SCM). La proporcionalidad del relleno de la celda está en función del número de actividades asociadas a la unidad de tiempo que se está correlacionando y la representación que se está utilizando (absoluta o relativa), además cada programador se encuentra representado mediante el uso de un color.

La Figura 2 muestra la línea de vida y las actividades que se han llevado a cabo sobre el archivo “macos.Delegate.java” entre el 2003 and 2008. Sin embargo, al revisar los paquetes que forman parte de la última revisión del sistema se puede notar que el paquete “macos” en la actualidad no forma parte esta, lo cual se puede corroborar al revisar la estructura del proyecto en Eclipse, como muestra la Figura 3(a). De acuerdo con lo anterior la línea de tiempo de un paquete o elemento de software está representada por la presencia de actividad en la fila correspondiente. Así, se puede distinguir su creación cuando aparece actividad asociada por primera vez y se puede deducir que la evolución del elemento se ha estabilizado, o bien este ha sido movido o eliminado cuando no se observan más actividades relacionadas con este. Mientras que el uso de los colores permite ver cuáles programadores han realizado cambios a un elemento y por lo tanto es posible conocer quiénes han colaborado con su evolución. Esta misma característica de la visualización permite establecer las relaciones socio-técnicas al relacionar de forma directa a los programadores con los elementos del sistema. Pero además también permite obtener detalles sobre las relaciones entre los programadores por los elementos que han cambiado en común, lo que resulta de utilidad a los administradores de proyectos para la asignación de tareas y la sustitución de un programador cuando sucede algún evento que lo aparta de forma temporal o permanente del proyecto.

5.4. Relaciones: herencia e implementación de interfaces

La estructura del sistema es una estructura plegable que permite expandir los paquetes para ver los archivos que contiene, y los archivos a su vez se pueden expandir para observar los elementos de software que contienen (ver Figura 3 (b)). Al continuar expandiendo la estructura, los elementos de software permiten ver con cuales otros elementos de software ha establecido en algún momento relaciones de herencia e implementación de interfaces. Las relaciones de herencia que han sido tomadas en cuenta consideran tanto a los ascendientes como a los descendientes (ver Figura 3 (c)).

La Figura 4 muestra que el establecimiento de las relaciones de herencia e implementación de interfaces es representada por un óvalo de color y que la terminación de esas relaciones es indicada por un óvalo de color rojo. De forma adicional también se indica de forma explícita la localización de los elementos de software asociados (API de Java, el sistema bajo análisis o una biblioteca externa).

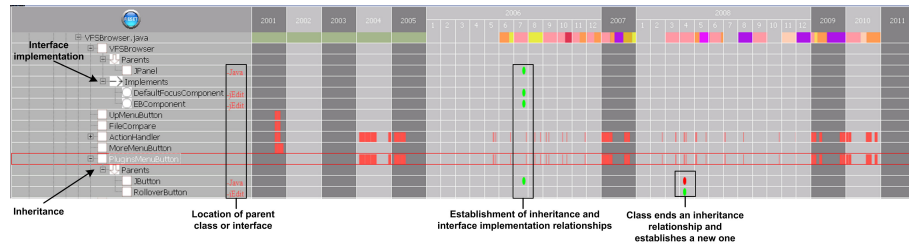


Figura 4. Relaciones de herencia e implementación de interfaces, incluyendo la expansión de un año y valores de métricas.

5.5. Métricas

Las métricas asociadas a los elementos de software son representadas haciendo uso de gráficas de barras con el fin de resaltar los cambios en sus valores (see Figure 5). De forma similar a la representación de las contribuciones de los programadores los valores de las métricas son mostrados usando áreas relativas y absolutas. La representación relativa utiliza el valor de la métrica más alto, tomando en cuenta todos los valores de las métricas asociadas a los elementos de software del sistema, y con base en ese valor calcula el alto de la gráfica, mientras que la representación absoluta solo toma en cuenta el valor de la métrica más alto para el elemento de software que está siendo analizado.

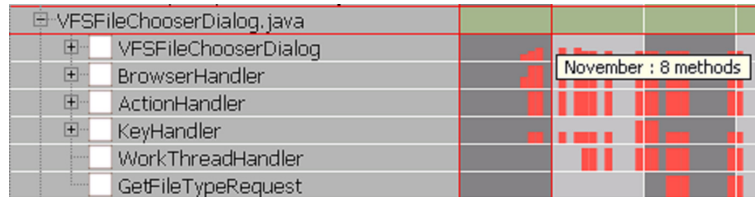


Figura 5. Representación de métricas.

5.6. Escenarios

Los siguientes escenarios buscan mostrar algunos ejemplos prácticos en los cuales GridMaster puede ser aplicada.

Escenario 1 Un error fue reportado y el administrador del proyecto debe asignar un programador para resolverlo. Una vez que el administrador del proyecto analizó el problema y determinó quien es el programador más indicado para resolverlo por estar a cargo de los componentes en los cuales es probable que se localice el error, se da cuenta que ese desarrollador se encuentra de vacaciones. Entonces el administrador del proyecto sigue los siguientes pasos utilizando GridMaster:

1. Abre la visualización mostrando el sistema completo, sin filtrar ningún paquete.
2. Selecciona una unidad de tiempo reciente que contiene un número considerable de actividades.
3. Revisa los elementos de sistema en los cuales considera que se puede encontrar localizado el error e inspecciona quienes han sido los programadores que han realizado cambios sobre esos elementos, con base en las relaciones socio-técnicas.
4. El administrador del proyecto determina cuál programador podría hacerse cargo de corregir el error y le asigna la tarea.

Escenario 2 Los programadores están en medio de una refactorización del sistema y necesitan revisar como la herencia y la implementación de interfaces ha sido afectada por los cambios realizados por los demás programadores. Por lo que los programadores siguen los siguientes pasos después de abrir GridMaster para revisar el sistema completo o solo un paquete que les resulta de su interés:

1. El programador selecciona un archivo, lo expande y selecciona el elemento de su interés.
2. Expande el elemento de software y examina las relaciones de herencia e implementación de interfaces para determinar si se ha establecido o terminado alguna de esas relaciones.

6. Conclusiones

El desarrollo, mantenimiento y evolución de sistemas de software en entornos distribuidos de forma global requiere de enfoques que contribuyan a mejorar la colaboración y la productividad de los equipos de trabajo. En esta investigación trabajo se ha discutido de forma amplia sobre la necesidad de la comunicación, la coordinación y el control, así como de la importancia de crear espacios que permitan desarrollar la conciencia situacional tanto de los individuos como de los equipos de trabajo. También se ha analizado el papel que juegan los procesos cognitivos individuales y grupales en la activación de los diferentes mecanismos que inician los procesos de colaboración.

De igual forma esta investigación discutió en profundidad el uso de la visualización para facilitar la comprensión de los sistemas de software y su evolución. Dicha discusión permitió resaltar la naturaleza dinámica de los sistemas de software, así como la dificultad para anticipar de forma precisa el tamaño y crecimiento de la arquitectura y estructuras subyacentes de estos. Lo que implica que el diseño de la representación visual de un sistema requiere ser escalable y capaz de acomodar tanto su crecimiento como los cambios en el tiempo (e.g. agregación, borrado y relocalización de elementos). Esto conlleva que la visualización debe ser consistente en el tiempo con el fin de mantener el mapa mental de los usuarios: los elementos deben aparecer en la misma posición de la visualización en todas las revisiones del sistema que sean representadas.

De forma adicional ha quedado en evidencia la importancia que tiene la representación de las métricas y relaciones entre elementos de software. Es común que el número de relaciones entre elementos se incremente de acuerdo al tamaño del sistema, lo que agrega un aspecto adicional al problema de diseño de visualizaciones escalables e incrementa considerablemente la dificultad para lograr un diseño adecuado. Lo que tiene mayores implicaciones cuando se efectúa la visualización completa o parcial de la evolución de un sistema. El aspecto temporal está ligado a la identificación de patrones relacionados con diferencias o similitudes entre revisiones o periodos de tiempo y requiere la representación de un mayor número de elementos visuales.

El análisis efectuado permitió reafirmar que la representación visual de un sistema (o parte de este) debe comprender elementos que permitan la rápida identificación de patrones de interés de acuerdo al problema que se busca resolver. Por lo que es necesario efectuar la representación en una sola pantalla y permitir que una vez que un patrón ha sido identificado se realice su revisión en una visualización adicional utilizando un enfoque de analítica visual.

Los resultados del análisis y la discusión efectuada condujo a la definición de un framework para describir la relación entre los elementos y factores relacionados con el desarrollo distribuido, el trabajo en equipo y la conciencia situacional. Esos resultados también contribuyeron con el diseño de una herramienta de visualización, denominada como GridMaster, que permite apoyar el desarrollo de la conciencia situacional y la colaboración entre los miembros del equipo de trabajo.

GridMaster no fue evaluada formalmente en esta investigación, pero su uso en escenarios concretos y situaciones controladas muestra su utilidad para proporcionar información sobre la arquitectura de los sistemas, las relaciones entre los elementos de la arquitectura, patrones de colaboración y las relaciones socio-técnicas.

La implementación de esta herramienta permitió mostrar la enorme utilidad que tiene el uso de la visualización para facilitar la comprensión de los sistemas de software y su evolución. Por lo que como trabajos futuros se plantea efectuar la evaluación formal de esta visualización con usuarios, así como realizar mejoras para permitir que los miembros del equipo de trabajo puedan hacer anotaciones cuando encuentran información relevante, y que además dichas anotaciones o descubrimiento de detalles puedan ser compartidas con los demás usuarios.

Glosario

AP	Administrador de Proyectos	4
CS	Conciencia Situacional.....	7, 11
CSET	Conciencia Situacional del Equipo de Trabajo.....	11
CVS	Concurrent Versioning System	10
DGS	Desarrollo Global de Software.....	1, 3, 4, 6, 11
DMES	Desarrollo, Mantenimiento y Evolución de Software ..	1-4, 6, 11-13
ECS	Espacio de Conciencia Situacional	10-14
ETCSD	Espacio de Trabajo de Conciencia Situacional Distribuida .	7
IDE	Entornos de Desarrollo Integrado	2, 3
SCM	Administración de la Configuración de Software .	3, 9, 14, 17

Bibliografía

- [1] J. Estublier, “Distributed objects for concurrent engineering,” in *System Configuration Management*, pp. 172–185, Springer, 1999.
- [2] J. D. Herbsleb and D. Moitra, “Global software development,” *Software, IEEE*, vol. 18, no. 2, pp. 16–20, 2001.
- [3] M. Jiménez, M. Piattini, and A. Vizcaíno, “Challenges and improvements in distributed software development: A systematic review,” *Advances in Software Engineering*, vol. 2009, p. 3, 2009.
- [4] M. Ogawa and K.-L. Ma, “code_swarm: A design study in organic software visualization,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 1097–1104, 2009.
- [5] E. Carmel, *Global software teams: collaborating across borders and time zones*. Prentice Hall PTR, 1999.
- [6] J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter, “An empirical study of global software development: distance and speed,” in *Proceedings of the 23rd international conference on software engineering*, pp. 81–90, IEEE Computer Society, 2001.
- [7] E. Ó. Conchúir, P. J. Ågerfalk, H. H. Olsson, and B. Fitzgerald, “Global software development: where are the benefits?,” *Communications of the ACM*, vol. 52, no. 8, pp. 127–131, 2009.
- [8] S. Misra, R. Colomo-Palacios, T. Pusatli, and P. Soto-Acosta, “A discussion on the role of people in global software development,” *Tehnicki vjesnik / Technical Gazette*, vol. 20, no. 3, pp. 525 – 531, 2013.
- [9] R. Prikladnicki, J. L. N. Audy, and R. Evaristo, “Global software development in practice lessons learned,” *Software Process: Improvement and Practice*, vol. 8, no. 4, pp. 267–281, 2003.
- [10] J. D. Herbsleb and A. Mockus, “An empirical study of speed and communication in globally distributed software development,” *Software Engineering, IEEE Transactions on*, vol. 29, no. 6, pp. 481–494, 2003.
- [11] I. Omoronyia, J. Ferguson, M. Roper, and M. Wood, “A review of awareness in distributed collaborative software engineering,” *Software: Practice and Experience*, vol. 40, no. 12, pp. 1107–1133, 2010.
- [12] A. Talaei-Khoei, P. Ray, N. Parameshwaran, and L. Lewis, “A framework for awareness maintenance,” *Journal of Network and Computer applications*, vol. 35, no. 1, pp. 199–210, 2012.
- [13] A. Mockus and D. M. Weiss, “Globalization by chunking: A quantitative approach,” *IEEE Software*, vol. 18, no. 2, pp. 30–37, 2001.
- [14] H. Mintzberg, “The effective organization: Forces and forms,” *Mit Sloan Management Review*, January, 15 1991.
- [15] D. W. Karolak, *Global Software Development: Managing Virtual Teams and Environments*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1st ed., 1999.

- [16] C. Ebert and P. De Neve, “Surviving global software development,” *Software, IEEE*, vol. 18, no. 2, pp. 62–69, 2001.
- [17] C. Ebert, C. H. Parro, R. Suttels, and H. Kolarczyk, “Improving validation activities in a global software development,” in *Proceedings of the 23rd international Conference on Software Engineering*, pp. 545–554, IEEE Computer Society, 2001.
- [18] D. Pinelle and C. Gutwin, “A groupware design framework for loosely coupled workgroups,” in *ECSCW 2005*, pp. 65–82, Springer, 2005.
- [19] S. L. Vrhovec, M. Trkman, A. Kumer, M. Krisper, and D. Vavpotič, “Outsourcing as an economic development tool in transition economies: Scattered global software development,” *Information Technology for Development*, no. ahead-of-print, pp. 1–15, 2014.
- [20] W. De Pauw, D. Kimelman, and J. Vlissides, “Modeling object-oriented program execution,” in *Object-Oriented Programming*, pp. 163–182, Springer, 1994.
- [21] D. Gračanin, K. Matković, and M. Eltoweissy, “Software visualization,” *Innovations in Systems and Software Engineering*, vol. 1, no. 2, pp. 221–230, 2005.
- [22] S. M. Fiore and E. Salas, “Why we need team cognition,” in *Team cognition: Understanding the factors that drive process and performance* (E. Salas and S. M. Fiore, eds.), pp. 235–248, American Psychological Association, 2004.
- [23] J. He, B. S. Butler, and W. R. King, “Team cognition: Development and evolution in software project teams,” *J. of Management Information Systems*, vol. 24, no. 2, pp. 261–292, 2007.
- [24] K.-P. L. Vu and R. W. Proctor, *Handbook of human factors in Web design*. CRC Press, 2011.
- [25] L. K. Comfort, “Crisis management in hindsight: Cognition, communication, coordination, and control,” *Public Administration Review*, vol. 67, no. s1, pp. 189–197, 2007.
- [26] Merriam-Webster Online, “Merriam-Webster Online Dictionary,” 2009.
- [27] M. M. Khan, S. A. Lodhi, and M. A. M. Makk, “Measuring team implicit coordination,” *Australian Journal of Basic and Applied Sciences*, vol. 4, no. 6, pp. 1211–1136, 2010.
- [28] J. Kotlarsky, P. C. Van Fenema, and L. P. Willcocks, “Developing a knowledge-based perspective on coordination: The case of global software projects,” *Information & Management*, vol. 45, no. 2, pp. 96–108, 2008.
- [29] P. Rook, “Controlling software projects,” *Software Engineering Journal*, vol. 1, no. 1, p. 7, 1986.
- [30] B. Ramesh, L. Cao, K. Mohan, and P. Xu, “Can distributed software development be agile?,” *Communications of the ACM*, vol. 49, no. 10, pp. 41–46, 2006.
- [31] M. R. Endsley, “Toward a theory of situation awareness in dynamic systems,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 32–64, 1995.
- [32] N. A. Stanton, P. R. Chambers, and J. Piggott, “Situational awareness and safety,” *Safety science*, vol. 39, no. 3, pp. 189–204, 2001.

- [33] P. M. Salmon and N. A. Stanton, "Situation awareness and safety: Contribution or confusion? situation awareness and safety editorial," *Safety science*, no. 56, pp. 1–5, 2013.
- [34] P. Leinonen, S. Järvelä, and P. Häkkinen, "Conceptualizing the awareness of collaboration: A qualitative study of a global virtual team," *Computer Supported Cooperative Work (CSCW)*, vol. 14, no. 4, pp. 301–322, 2005.
- [35] N. A. Stanton, R. Stewart, D. Harris, R. J. Houghton, C. Baber, R. McMaster, P. Salmon, G. Hoyle, G. Walker, M. S. Young, *et al.*, "Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology," *Ergonomics*, vol. 49, no. 12-13, pp. 1288–1311, 2006.
- [36] R. Kazman, G. Abowd, L. Bass, and P. Clements, "Scenario-based analysis of software architecture," *Software, IEEE*, vol. 13, no. 6, pp. 47–55, 1996.
- [37] T. Khan, H. Barthel, A. Ebert, and P. Liggesmeyer, "Visualization and evolution of software architectures.," in *VLUDS*, pp. 25–42, 2011.
- [38] Y. Assogba and J. Donath, "Share: a programming environment for loosely bound cooperation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 961–970, ACM, 2010.
- [39] A. Jermakovics, A. Sillitti, and G. Succi, "Mining and visualizing developer networks from version control systems," in *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering*, pp. 24–31, ACM, 2011.
- [40] B. Heller, E. Marschner, E. Rosenfeld, and J. Heer, "Visualizing collaboration and influence in the open-source software community," in *Proceedings of the 8th Working Conference on Mining Software Repositories*, pp. 223–226, ACM, 2011.
- [41] M. Lanza, L. Hattori, and A. Guzzi, "Supporting collaboration awareness with real-time visualization of development activity," in *Software Maintenance and Reengineering (CSMR), 2010 14th European Conference on*, pp. 202–211, IEEE, 2010.
- [42] T. Girba, A. Kuhn, M. Seeberger, and S. Ducasse, "How developers drive software evolution," in *Principles of Software Evolution, Eighth International Workshop on*, pp. 113–122, IEEE, 2005.
- [43] L. Hattori, M. Lanza, and R. Robbes, "Refining code ownership with synchronous changes," *Empirical Software Engineering*, vol. 17, no. 4-5, pp. 467–499, 2012.
- [44] E. Salas, C. Prince, D. P. Baker, and L. Shrestha, "Situation awareness in team performance: Implications for measurement and training," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 123–136, 1995.
- [45] P. Young and M. Munro, "Visualising software in virtual reality," in *Program Comprehension, 1998. IWPC'98. Proceedings., 6th International Workshop on*, pp. 19–26, IEEE, 1998.
- [46] J. Maletic, M. L. Collard, *et al.*, "Supporting source code difference analysis," in *Software Maintenance, 2004. Proceedings. 20th IEEE International Conference on*, pp. 210–219, IEEE, 2004.

Tools for Teaching Logic

Juan Carlos Uclés Mármol¹ y María Gracia Manzano Arjona²

¹ Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España

`jcucles@usal.es`

² Departamento de Filosofía, Lógica y Estética, Universidad de Salamanca. Campus Miguel de Unamuno, 3. 37007 Salamanca, España

`mara@usal.es`

Resumen Three International Congresses were celebrated in Salamanca in 2000, 2006 and 2011 about tools to teach logic. This report will focus exclusively on software tools, leaving aside other pedagogical resources that were presented during the conferences. It aims to give an update about these computer tools presented in the congresses and verify its current performance. It will also show the evolution of the technological platforms used, beginning with desktop tools and moving on to E-Learning platforms. This report will show the nowadays resources that logic counts, in terms of software application for teaching basic university level. *abstract* environment.

Keywords: Logic, Tools, Software, E-learning, Congress, Platforms, Interactive

1. Introducción

Bajo la denominación *Tools for Teaching Logic* se han organizado tres congresos de carácter internacional en la Universidad de Salamanca. Según declararon sus organizadores.

El objetivo principal de este congreso será el conseguir que un elevado número de investigadores de alto nivel científico compartan no sólo su pasión por la investigación sino también por la enseñanza de la lógica a todos los niveles educativos, tanto los de grado como los de posgrado. Pretendemos mostrar cómo la investigación básica y la aplicada pueden y deben estar ligadas a la innovación docente.¹

La iniciativa parte de un proyecto ALFA (América Latina Formación Académica) de la Unión Europea que coordinó Salamanca durante 1998-1999 y que permitió crear la red ARACNE (<http://aracne.usal.es/index.html>). Participaron en este proyecto 4 países europeos y cinco de Iberoamérica: Argentina, Brasil,

¹ Documento de Memoria técnica de Acciones Complementarias Modalidad A (Organización de congresos, seminarios, simposios, reuniones y jornadas de carácter científico-técnico).

España, Holanda, Italia, México, Perú, Reino Unido y Uruguay. De entre los logros de este proyecto destacaría la biblioteca digital (<http://logicae.usal.es/>) que tan útil resulta a los estudiosos del tema. El proyecto finalizó en el 2000 con un encuentro internacional, el First International Congress on Tools for Teaching Logic (<http://aracne.usal.es/congress/congress.html>).

En el año 2006 se realizó un segundo congreso (<http://logicae.usal.es/SICTTL>), que superó al anterior tanto por la variedad de los temas abordados, como por la diversidad de países participantes y por la calidad de las ponencias. Los numerosos profesionales que asistieron provenían de 40 universidades pertenecientes a 17 países de varios continentes. En Junio de 2011 se realizó el tercero de estos congresos en Salamanca (<http://logicae.usal.es/TICTTL/>) cuya nota más destacada fue la variada participación internacional y la videoconferencia de Stephen Wolfram, el creador del famoso software **Mathematica**.

En la actualidad las descargas sobre los artículos del último congreso ha sido muy alta con un aproximado de 4.339 capítulos bajados hasta la primavera del 2013 del volumen de actas publicado por Springer.

1.1. Proceso Investigativo

Inicié mi investigación analizando dichos congresos. He tenido acceso a las actas de los mismos que aparecen publicadas en las correspondientes páginas web y también a las publicaciones selectas basadas en las conferencias de los congresos y publicadas en la revista internacional IGPL [1] (las del segundo congreso) y en un libro de Springer, *Tools for Teaching Logic, Lecture Notes in Computer Science*, Vol. 6680 (las del tercer congreso). El resultado de este trabajo pretendo presentarlo al cuarto *Tools for Teaching Logic*, que se celebrará en 2015 en Rennes, Francia (<http://ttl2015.irisa.fr/>).

En este trabajo fin de máster he realizado una recopilación, junto con un seguimiento y un análisis de una serie de herramientas presentadas en los tres congresos internacionales de lógica denominados *Tools for Teaching Logic* (años 2000, 2006 y 2011), que son los únicos que con estas características creo existen a nivel mundial.

Considero importante mencionar que no existía una investigación similar a la que presento y además he mantenido contacto con muchos de los investigadores o desarrolladores de las herramientas software para conocer el estado actual de las mismas, es decir si han sufrido una evolución, continúan en uso o han quedado obsoletas.

Como su nombre lo indica, las herramientas presentadas en los congresos, son para la enseñanza de la lógica en los diversos cursos que se imparten de la misma en los grados de ciencias de computación, matemáticas y filosofía. Es importante señalar que ellos están entendiendo como Herramienta, a cualquier metodología utilizada para la enseñanza de la lógica y que en este trabajo he procurado centrarme en una parte de ellas, las herramientas o aplicaciones informáticas.

1.2. Fundamentos Teóricos

Definiendo la Lógica, al analizar esta ciencia desde la perspectiva actual, la lógica es la materia interdisciplinar por excelencia ya que actúa como núcleo de una ciencia que emerge: la de la transmisión de la información. Por consiguiente, concentrarnos en crear material educativo para la enseñanza de la lógica tiene un carácter ejemplificador pues en ella se funden disciplinas en donde son determinantes los aspectos simbólicos del proceso de información [2].

Relacion entre Lógica e Informática, en informática la lógica formal es el principal soporte matemático utilizado tanto en el diseño de hardware, como en los desarrollos de los lenguajes de programación y en la construcción de la Inteligencia Artificial.

En lo relativo al hardware, es la base teórica, tanto la de los ordenadores digitales, como en los que lo hacen con circuitos integrados que utilizan lógica de algún tipo en su diseño. Sucede algo similar con los lenguajes de programación que no son más que lenguajes formales cuyas fórmulas bien formadas son los programas; en inteligencia artificial usamos la lógica para realizar cálculos mediante razonamiento automático al utilizar un sistema experto.

La Informática ha estado, desde sus inicios, fuertemente inspirada e influida técnicamente por la Lógica. Tal es el caso de muchos lógicos notables como Turing y Church los que tienen también un nombre importante en la historia de la Informática y cuyas aportaciones precedieron al nacimiento de la informática.

En la actualidad esta gran influencia entre lógica e informática perdura, demostrándose por ejemplo en las Lógicas de Programas, y en los Lenguajes de Programación Lógica [2, 3].

El interés de la informática por la lógica es creciente, como se puede constatar en los numerosos handbooks para inteligencia artificial, programación y Computación [4, 5, 6].

Aprendizaje Visual y Lógica, Los psicólogos han estado interesados en la relación entre la visualización y el mecanismo de razonamiento humano. Los matemáticos, por ejemplo, han apreciado el valor de los diagramas y otras herramientas visuales tanto para la enseñanza como para el proceso de descubrimiento matemático. La representación visual es de vital importancia para comprender mejor los conceptos que se deben aprender en la lógica; los diagramas en geometría, por ejemplo, son el mejor camino para probar los teoremas geométricos.

En cuanto al uso de la información extra lingüística, existen pruebas suficientes para suponer que gran parte, por no decir que la mayoría, del razonamiento hace uso de alguna forma de representación visual para su comprensión. Los diagramas servirán para resolver una clase muy amplia de problemas, incluyendo los razonamientos abstractos.

Una de las mayores ventajas de un diagrama es que ilustra muy bien lo representado a lo largo de muchas dimensiones, lo suficientemente efectivos para

que una persona pueda, por decirlo de alguna forma, *leer* los hechos sobre el objeto representado.

A medida que la ciencia de la computación nos proporcione más herramientas visuales para representar la información, tendremos que empezar a estudiar los aspectos lógicos del razonamiento que utilizan formas no lingüísticas de representación (Diagramas).

Lo que se espera es que las herramientas informáticas que se han empezado a utilizar permitan que la información representada tenga más de un modo de visualización, con lo que generará mayor entendimiento o comprensión de lo que se enseña en las clases de Lógica. (Conclusiones del libro, Logical Reasoning with Diagrams [7])

E-learning y Uso de Internet Una de las aplicaciones más interesantes de Internet es su uso como una poderosa herramienta de aprendizaje, así como la constitución de cualquier tipo de comunidades (científica, de investigación, de aprendizaje) constituida por individuos permanentemente informados y actualizados entre sí. La aplicación de herramientas basadas en la Web con fines de aprendizaje podría ser entendida como una simple definición del termino e-Learning.

E-Learning se podría definir como un proceso de enseñanza-aprendizaje con el objetivo de obtener un conjunto de habilidades y competencias de los estudiantes, tratando de garantizar la máxima calidad en todo el proceso, gracias a:

- * Principalmente, el uso de tecnologías basadas en la Web.
- * Contenidos bien estructurados sobre la base de estrategias predefinidas pero flexibles.
- * Interacción con el resto del grupo formado por estudiantes y profesores.
- * Procedimientos de evaluación adecuados, tanto de los resultados del aprendizaje como del propio proceso de enseñanza en conjunto.
- * Entorno de trabajo de colaboración con espacio-tiempo diferenciando al presencial.
- * Una suma de servicios tecnológicos de valor añadido con el fin de lograr la máxima interacción.

1.3. Resumen Cuantitativo de Artículos Analizados

A continuación se describirán las herramientas informáticas que se han presentado en los distintos congresos de lógica; la mayoría de dichas herramientas cuentan con una interfaz gráfica que auxilia a los usuarios (estudiantes) en la realización de las tareas encomendadas en los cursos de lógica elemental. Se estudiaron en detalle los artículos presentados, clasificándolos por congreso: I congreso (junio del año 2000), II congreso (Septiembre del año 2006) y III congreso (junio del año 2011).

Se modela en la figura 1, el porcentaje de las herramientas informáticas presentadas en los tres congresos internacionales de lógica, con lo cual podemos observar el incremento de este tipo de artículos referentes a plataformas web o

aplicaciones para la enseñanza de la lógica, respecto de otro tipo de herramientas.

Herramientas Informáticas Presentadas

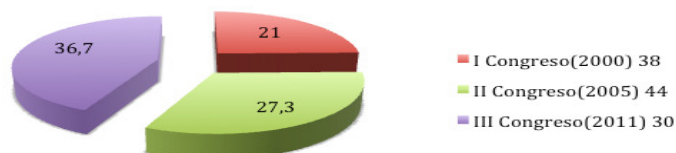


Figura 1. Gráfico porcentual de las herramientas software presentadas por congreso.

1.4. Listado de Herramientas Informáticas Analizadas

A continuación se presenta un listado de las herramientas informáticas analizadas para cada uno de los tres congresos, Tools for Teaching Logic. Como se

Tools for Teaching Logic I	Tools for Teaching Logic II	Tools for Teaching Logic III
Visual Da Vinci	Pandora	Araucaria-PL
WinKE	The Web ITP Tool	ORGANON
Smullyan	ORGANON	SATOULOUSE
Dinamul	LoTREC	PANDA
ADN (Asistente Para Deducción Natural)	MAFIA	The SELL Project
Mathematical Logic Tutor	Tool for manipulating logical formulae(Ideas)	A teaching Tool for Proving Equivalences(Ideas)
Procesamiento del Lenguaje Natural(Prolog)	Morph Moulder(MoMo):	OVA
SIAL: Intelligent Tutoring System	The Athena Logic Software	ARVINA
	Visual ITP	Parley
	EasyLogic	Logicamente
	Modelos de Kripke	The AProS Project
	TARSKI'S WORLD 5.0	

Figura 2. Cuadro Resumen de Herramientas Analizadas.

puede observar en la figura 2, desde el principio de la iniciativa en el año 2000

cuando tuvo lugar el Primer Congreso, se han presentado (En total 31 Aplicaciones) herramientas software que han sido de utilidad en el mundo de la lógica; el estado de algunas de estas herramientas de enseñanza en la actualidad se mencionará en la sección(5).

2. Descripción Primer Congreso *Tools for Teaching Logic* (año 2000)

El logro principal de este congreso de carácter internacional fue el de reunir a un elevado número de investigadores de alto nivel científico para que compartieran sus avances en cuanto a herramientas pedagógicas para la enseñanza de la lógica. El número de participantes o investigadores así como la variedad de los países tanto europeos como norteamericanos y de habla hispana es destacable.

Se realizaron diversas conferencias, sesiones de debate, mesas redondas y demostraciones de software. Hubo varias sesiones en las que de forma activa se mostraban las herramientas software utilizadas en la enseñanza de la lógica.

Temas Principales:

- 1º Cuestiones y objetivos en la enseñanza de la lógica.
- 2º La enseñanza de la lógica en Ciencias y Humanidades..
- 3º Una aproximación epistemológica (Cognoscitiva) a la enseñanza de Lógica.
- 4º Enseñanza del razonamiento y las demostraciones en Lógica y Matemática en el bachillerato.
- 5º Enseñanza Lógica Utilizando la Web (Recursos y Desafíos).
- 6º Software Educativo y Profesional de Lógica.

Página Web para Mayor Información: <http://aracne.usal.es/congress/congress.html>, información disponible en el libro de actas para el primer congreso [8].

Herramientas Informáticas Presentadas: Primer Congreso Se puede observar en la tabla 1 que las herramientas presentadas son en total 8. En esta ocasión, las aplicaciones presentadas provenían de 4 países (España, Argentina, Alemania, Inglaterra)

Herramienta Presentada	Tipo de Aplicación
Visual DaVinci	Escritorio
WinKE	Escritorio
Smullyan	Escritorio
Dinamul	Escritorio
ADN (Asistente Para Deducción Natural)	Escritorio
Mathematical Logic Tutor	Escritorio
Procesamiento del Lenguaje Natural(Prolog)	Escritorio

SIAL: Intelligent Tutoring System	Escritorio
-----------------------------------	------------

Cuadro 1: Herramientas Informáticas Primer Congreso

Sobre las herramientas del Primer Congreso. Se puede indicar que en el Congreso realizado en el año 2000, solo han sobrevivido unas pocas herramientas como son: Visual Da Vinci, WinKE y SIAL(SLI) las cuales son aplicaciones de escritorio y sirven para mejorar el manejo de cálculos deductivos por parte de los alumnos que toman cursos básicos de Lógica e Inteligencia Artificial.

Como ejemplo, la interfaz de WinKE(Figura 3), para construir un árbol de pruebas(es como usar lápiz y papel, visualmente satisfactorio, facilidad de uso, calidad Latex).

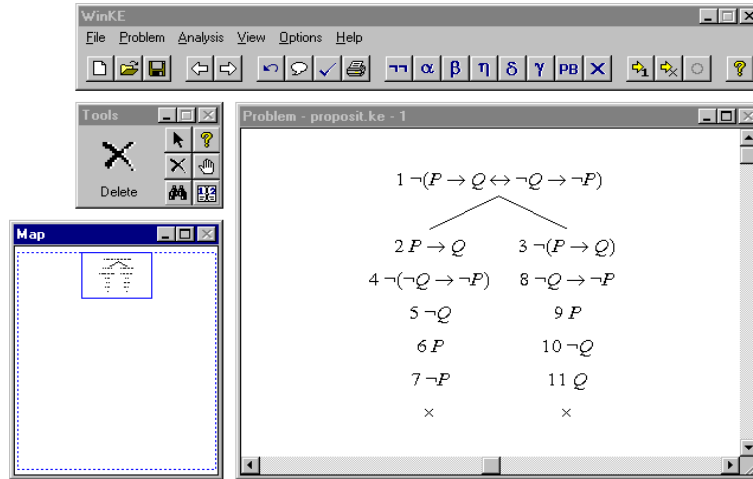


Figura 3. Interfaz WinKE.

3. Descripción Segundo Congreso *Tools for Teaching Logic* (año 2006)

Los participantes de este segundo congreso manifiestan compartir la opinión de que la enseñanza de una materia tan interdisciplinar, que abarca el campo de la lógica, la lingüística y las ciencias de la computación, debería estar disponible para la comunidad universitaria de tal manera que se facilite la investigación en estos campos.

No obstante, aseguran ser conscientes de que las necesidades son diferentes en los distintos campos de estudio que ya han sido mencionados. La preocu-

pación general es en la enseñanza de la lógica, pero con especial atención en el tratamiento de las innovaciones y en la sistematización de la actividad educativa.

Creemos que el papel de la lógica en la conformación de la epistemología de este siglo XXI debería ser crucial; las tecnologías de la información están cambiando rápidamente el mundo en que vivimos, y la lógica nos está ayudando a producir, distribuir y procesar la información; así mismo la lógica nos sirve para entender cómo la información codificada puede modificar los estado de conocimiento de los agentes implicados en el proceso de adquisición y distribución de la información.

Se sostuvieron charlas, sesiones de debate, mesas redondas y demostraciones de software, donde se trataron los siguientes temas:

- 1º Problemas, medios y objetivos en la enseñanza de la lógica..
- 2º La enseñanza de la lógica en Ciencias y Humanidades.
- 3º Enseñanza de la Lógica en los Programas de Postgrado.
- 4º Software para la Enseñanza de la lógica y del razonamiento.
- 5º E-learning en Lógica: Recursos y desafíos.

Página Web para Mayor Información: <http://logicae.usal.es/SICTTL/>, información disponible en el libro de actas para el primer congreso [9].

Herramientas Informáticas Presentadas: Segundo Congreso. Se puede observar en la tabla 2 que las herramientas presentadas son en total 12. En esta ocasión los desarrolladores eran la mayoría europeos, de 8 países (España, Alemania, Inglaterra, Holanda, República Checa, Suecia, Estados Unidos, Francia)

Herramienta Presentada	Tipo de Aplicación
Pandora	Escritorio y Web
The Web ITP Tool	Web
ORGANON	Web
LoTREC	Escritorio
MAFIA	Escritorio
Tool for manipulating logical formulae(Ideas)	Web
Morph Moulder(MoMo)	Escritorio
The Athena Logic Software	Escritorio
Visual ITP	Escritorio
EasyLogic	Web
Modelos de Kripke	Escritorio
TARSKI'S WORLD 5.0	Escritorio

Cuadro 2: Herramientas Informáticas Segundo Congreso

En esta ocasión, los organizadores del segundo congreso publicaron algunos de sus resultados en la revista *Logic Journal of the IGPL* (Interest Group in Pure and Applied Logics) [1]. Se trata de una revista de gran impacto, que ocupa la primaria posición entre las revistas de Lógica ²

En este número de IGPL aparecieron sólo algunos artículos seleccionados por los organizadores y revisados por la propia la revista; entre ellos los referentes a: Pandora, ORGANON, Morph Moulder(MoMo), Mafia, Modelos de Kripke, además de artículos relacionados con las opciones de educación en línea a través de plataformas virtuales. Uno de ellos, **Teaching and Learning Logic in a Virtual Learning Environment** (Antonia Huertas) es el primero de la lista de los que mayor éxito han tenido (descargas) de toda la revista *Articles* (http://oxfordjournals.org/our_journals/igpl/topd.html)

Con respecto a las herramientas del Segundo Congreso. En esta etapa sobresale la aparición de nuevas aplicaciones basadas en plataformas Web como son: EasyLogic, Tool for manipulating logical formulae (Ideas), Organon, que se convierten en los primeros tutores en línea para la enseñanza de la Lógica. Estos programas permiten la interacción y manipulación de formulas (ejercicios) a los estudiantes y ayudan a los profesores a reducir la carga (reducción de consultas y asuntos administrativos por parte de los alumnos).

No dejan de ser menos importantes otras herramientas de escritorio como son: Pandora, LoTREC 2.0, Morph Moulder (MoMo), TARSKI'S WORLD 5.0, las que funcionan de forma eficaz y potente, como material de apoyo en clase para la comprensión de conceptos abstractos de Lógica.

Podemos observar como ejemplo en la figura 4 la interfaz de TARSKI'S WORLD, que sirve para la comprensión de reglas semánticas.

4. Descripción Tercer Congreso *Tools for Teaching Logic* (año 2011)

Quiero señalar que aunque no se trata de un congreso de carácter periódico, fue ésta la tercera vez que se logró organizar el mismo, y se presentaron nuevas herramientas pedagógicas para la enseñanza de la lógica.

La lógica es abordada como materia de estudio e investigación en las ramas de filosofía, matemáticas e informática y con frecuencia también en la de psicología. Por esta razón los ponentes invitados fueron filósofos, matemáticos e informáticos de distintos países de Europa y de América, incluso de Nueva Zelanda. En todos los casos se trata de investigadores de primera línea.

Esta vez se contó con el apoyo institucional de **La Sociedad de Lógica, Metodología y Filosofía de la Ciencia en España** <http://www.solofici.org/>, **(FoLLI) Association for Logic, Language and Information**(<http://www.folli.info>)

² Journal of the IGPL is delighted to announce that it is still ranked Number 1 in the Logic category and has been awarded an Impact Factor of 1.136 for 2012. 2012 Citation Reports " (Thomson Reuters, 2012)

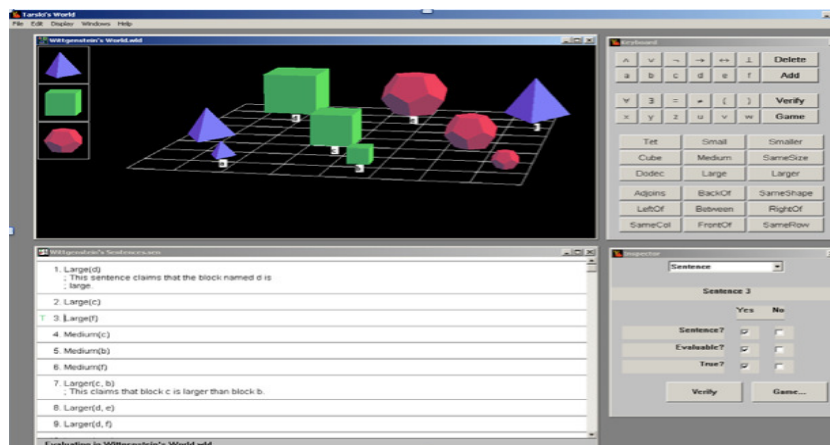


Figura 4. Interfaz Tarski's World.

y la **Association for Symbolic Logic** <http://www.aslonline.org/index.htm>. Se aprovechó para gestionar la participación de otras instituciones del mundo iberoamericano.

El congreso se centró en una variedad de temas, incluyendo:

- 1º Software para la enseñanza de la lógica.
- 2º Métodos formales en la enseñanza.
- 3º La lógica de las humanidades.
- 4º La difusión de material didáctico y libros de texto de lógica.
- 5º E-learning en Lógica: Recursos y desafíos.
- 6º Los métodos para la enseñanza de la lógica en los diferentes niveles de enseñanza (Bachillerato, Universitaria y de Posgrado).
- 7º La presentación de programas de postgrado en lógica.
- 8º E-learning y juegos de lógica.
- 9º La enseñanza de la teoría de la argumentación y de la lógica informal.
- 10º La pedagogía de la lógica.

Algunas de las contribuciones del congreso formaron parte de las actas impresas, publicadas en Springer (<http://link.springer.com/book/10.1007/978-3-642-21350-2>) y que estuvieron disponibles el primer día del Congreso. El resto de las comunicaciones están disponibles en línea en la página del congreso. Todos los resúmenes de todas las comunicaciones están disponibles on-line. Desde su publicación en línea desde del año 2011 hasta el año 2013 suman un total de 4.339 descargas de capítulos de este libro en SpringerLink. La Tabla 3 muestra las cifras de descargas de los últimos años.

Año	Capítulos Descargados
2013	2939
2012	680

2011	720
------	-----

Cuadro 3: Descarga de Capítulos por Año

Página Web para Mayor Información: <http://logicae.usal.es/TICTTL/index.html>, información disponible en el libro de actas para el tercer congreso [10]

Herramientas Informáticas Presentadas: Tercer Congreso. Como se puede observar en la tabla 4, las herramientas presentadas fueron en total 11. En esta ocasión fueron 8 los países que presentaron aplicaciones (Alemania, Holanda, República Checa, Brasil, Estados Unidos, Francia, Escocia, Guatemala)

Herramienta Presentada	Tipo de Aplicación
Araucaria-PL	Web
ORGANON	Web
SATOULOUSE	Escritorio
PANDA	Escritorio
The SELL Project	Web
Exercise Assistant Online	Web
OVA	Web
ARVINA	Web
Visual ITP	Escritorio
Parley	Web
Logicamente	Web
The AProS Project	Web

Cuadro 4: Herramientas Informáticas Tercer Congreso

Mencionando algunas herramientas del Tercer Congreso. Hemos observado que en esta ocasión la mayoría de las aplicaciones están aun vigentes e incluso han evolucionado hacia plataformas en línea (Web); aunque se podría hacer mención de casi todas ellas. Por el grado de funcionalidades y sus fortalezas nos referiremos a: The SELL Project, The AProS Project y ORGANON, que son ahora mismo tutores en línea bastante potentes y ofrecen metodologías de aprendizaje auto-guiadas a los alumnos, guardan registro de actividad, realizan y revisan pruebas en línea, asignan tareas, generan datos estadísticos entre otros. Además, sirven de apoyo a los profesores o tutores disminuyendo tareas en cuanto a consultas triviales y/o cargas administrativas.

Como ejemplo en la Figura 5, la interfaz de The SELL Project, es un tutor en línea utilizado para el aprendizaje de calculo proposicional y de predicados (Primer Orden).

Se aborda también el tópico de la teoría de la argumentación en donde desde hace muy poco se cuenta con aplicaciones informáticas diseñadas para su análisis,

Castellano · Català · English

UOC Asistente e-Learning Lògica
v.5 beta

[Inicio](#) | [Lógica de Enunciados: Formalización - Deducción natural - Resolución - Tablas de verdad](#) | [Lógica de Predicados: Deducción natural - Resolución - Fórmulas a Enunciados](#) | [Salir](#)

Usuario: Juan Uclés

Lógica de Enunciados » Deducción natural

Ejercicios propuestos recomendados Ver todos

N.	Ejercicio	R	Resolver
1.	Ejemplo de introducción de la negación $T \rightarrow (W \rightarrow Z)$ $\neg Z \wedge W$ $\therefore \neg T$	<input type="checkbox"/>	Resolver
2.	Ejemplo de introducción de la implicación Q $R \wedge S \rightarrow T$ $P \rightarrow R$ $Q \rightarrow S$ $\therefore P \rightarrow T$	<input type="checkbox"/>	Resolver
3.	Ejemplo de eliminación de la disyunción $(A \rightarrow C) \vee (A \rightarrow D)$ $\therefore A \rightarrow C \vee D$	<input type="checkbox"/>	Resolver
4.	Ejemplo de introducción de la negación $P \rightarrow Q$ $T \rightarrow Q \rightarrow W$ $P \rightarrow T$	<input type="checkbox"/>	Resolver

Mis enunciados

Resumen

Ejercicios propuestos		Ejercicios propios	
Completos	0	Completos	0
Incompletos	0	Incompletos	0
Pendientes	28		

[Ver estadísticas](#)

Mis ejercicios [Crear](#)

Figura 5. Interfaz The Sell Project.

como son: Araucaria-PL, OVA, Arvina y Parley. Dichas herramientas hacen uso de algoritmos específicos para facilitar el análisis de los argumentos así como su manipulación sobre entornos en línea, con un enfoque hacia las visualizaciones esquemáticas.

Podemos observar como demostrativo en la figura 6 la interfaz de Ova, para el tratamiento de argumentos en línea.

The image shows a web browser window displaying a news article on the left and an argument diagram on the right. The news article is from BBC and discusses a power line upgrade in Scotland. The argument diagram on the right is a flowchart with nodes containing text fragments from the article, such as 'The line should be built', 'the need to provide more elect...', and 'the potential environmental dis...'. The diagram uses arrows to show the logical relationships between these statements.

Figura 6. Interfaz Ova.

5. Conclusiones

1.- Señalamos la importancia de la lógica para el ser humano; se trata de un arma poderosa que ha estado siempre presente en las distintas ciencias desarrolladas por la humanidad a lo largo de los siglos, Así como en los negocios, la ingeniería, la filosofía, matemática, etc. Ya que es a través de ella que los humanos podemos obtener razonamientos efectivos sobre situaciones o problemas que de otra forma sería difícil resolver. Así lo hemos constatado en el caso de las teorías de la argumentación y en el de las inferencias lógicas logradas a través de la aplicación de técnicas o herramientas que nos ayudan a comprender nuestro entorno.

2.- En este trabajo hemos podido observar la evolución de las herramientas presentadas desde el Primer Congreso *Tools for Teaching Logic* hasta la actualidad. Hemos verificado que las herramientas informáticas presentadas desde el año 2000 hasta el presente han sufrido grandes cambios, para beneficio de la lógica y su enseñanza, por lo que se podría predecir que en los próximos años se producirán también variaciones importantes y que el uso de plataformas informáticas interactivas para la enseñanza de la Lógica se incrementara. Su uso e introducción se extenderá a las distintas Universidades que imparten este tipo de cursos. Por lo que su implementación constituye un reto y una gran oportunidad para las nuevas tecnologías de la información y la comunicación.

Bibliografía

- [1] H. van Ditmarsch and M. Manzano, “Editorial ‘tools for teaching logic’,” *Logic Journal of IGPL*, vol. 15, no. 4, pp. 289–292, 2007.
- [2] M. Manzano, “‘¿qué es esa cosa llamada lógica?’,” *Summa logicae en el siglo XXI.M.*, 2005.
- [3] M. Manzano, “La bella y la bestia (perdón, lógica e informática),” *Summa logicae en el siglo XXI.M.*, 2002.
- [4] D. M. Gabbay, C. J. Hogger, and J. A. Robinson, eds., *Handbook of Logic in Artificial Intelligence and Logic Programming (Vol. 1)*. New York, NY, USA: Oxford University Press, Inc., 1993.
- [5] S. Abramsky, D. M. Gabbay, and S. E. Maibaum, eds., *Handbook of Logic in Computer Science (Vol. 2): Background: Computational Structures*. New York, NY, USA: Oxford University Press, Inc., 1992.
- [6] J. v. Eijck and H. Kamp, “Discourse Representation in Context,” in *Handbook of Logic and Language* (J. v. Benthem and A. ter Meulen, eds.), Elsevier, 2010.
- [7] T. Hailperin, “G. allwein and j. barwise (eds.), logical reasoning with diagrams,” *History and Philosophy of Logic*, vol. 17, pp. 169–169, 1996.
- [8] M. Manzano, “Libro de actas, first international congress on tools for teaching logic,” 2000.
- [9] B. P. L. M. Manzano and A. Gil, “Libro de actas, second international congress on tools for teaching logic,” 2006.
- [10] P. Blackburn, H. van Ditmarsch, F. Soler-Toscano, and M. Manzano, eds., *TICTTL’11: Proceedings of the Third International Congress Conference on Tools for Teaching Logic*, (Berlin, Heidelberg), Springer-Verlag, 2011.

Organizaciones virtuales de agentes para la integración de personas discapacitadas

Alberto López Barriuso¹ y Juan Manuel Corchado Rodríguez¹

¹Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España
{albarriuso, corchado}@usal.es

Resumen En este trabajo se realiza el diseño e implementación de una arquitectura que permite modelar la actividad de un entorno laboral de oficina, con el objetivo de identificar las barreras arquitectónicas que pueda presentar y dificulten la accesibilidad y los procesos del ámbito laboral. Podremos mejorar de esta manera la accesibilidad del entorno de trabajo a personas discapacitadas. Así, mediante la incorporación de mecanismos y técnicas de simulación es posible encontrar áreas susceptibles de mejora. Finalmente, como complemento se presenta también un modelo de visualización del entorno simulado, que facilita la toma de decisiones sobre la mejora de procesos de trabajo que ocurran en el entorno. Con tal fin, el estado del arte muestra los Sistemas Multi-Agente basados en Organizaciones Virtuales de Agentes y los modelos de Simulación Social Basada en agentes como la base tecnológica necesaria para hacer frente a los problemas a resolver.

Keywords: Agente Inteligente, Sistema Multi-Agente, Organización Virtual, Simulación Social Basada en Agentes

1. Introducción

Hay ciertos colectivos que se enfrentan a desventajas claras, uno de estos grupos es el de los discapacitados. Según estudios realizados por Naciones Unidas [1], en torno al 15% de la población mundial sufre algún tipo de discapacidad, así de acuerdo con *International Labour Organization* (ILO), existen alrededor de 386 millones de personas en edad de trabajar que sufren alguna discapacidad, llegando la tasa de desempleo entre éstos al 80% en el caso de algunos países.

Por ello, es necesario tomar las medidas adecuadas para intervenir de manera que se mejore y favorezca su integración social, y por extensión, ocupacional. Ante esta problemática, existe una creciente preocupación por parte de los distintos gobiernos, quienes promueven la integración laboral de los discapacitados a través de distintas normativas y directivas, por ejemplo las *Normas Uniformes sobre la igualdad de oportunidades para las personas con discapacidad* [2].

La falta de accesibilidad de los edificios y la integración de los discapacitados en el mundo laboral son dos de los elementos clave sobre los que se asientan las distintas normativas. Por norma general, los gobiernos obligan a las empresas

con un determinado número de trabajadores a la contratación de un cierto número de personas discapacitadas, por lo que los edificios donde estas empresas desarrollan su actividad, deben estar adaptados a las necesidades de estos colectivos, facilitando la accesibilidad a las personas discapacitadas según la legislación vigente.

Este estudio se centra en la adecuación de entornos laborales, haciendo posible determinar los problemas arquitectónicos de los éstos (principalmente entornos de oficina), lo que facilitará su adecuación y accesibilidad con la eliminación de obstáculos, reduciendo así el aislamiento y dependencia de los discapacitados y ayudando a validar el modelo del edificio ante distintos problemas de accesibilidad antes de su construcción.

Con este objetivo, se realizará una investigación y posterior desarrollo de un sistema multi-agente basado en organizaciones virtuales dirigido a la integración laboral de personas con discapacidades motrices. El sistema modelará el entorno de trabajo mediante organizaciones virtuales, de forma que los agentes adquieran los roles de los actores que intervienen en el entorno de trabajo, además de definir la distribución de los elementos dentro del edificio y planificar las tareas que los trabajadores realicen dentro de éste.

Gracias a sistema seremos capaces de monitorizar la interacción entre los agentes, con la finalidad de extraer, tras la realización de simulaciones, los distintos problemas de accesibilidad a los que los trabajadores se hayan podido enfrentar.

2. Objetivos

Gracias a los avances tecnológicos, podemos enfocar nuevas soluciones a problemas que se presenten, pero para ello debemos prestar atención a la hora de elegir la tecnología y abstracción a utilizar para lograr tales objetivos.

Como hemos introducido, la falta de accesibilidad de los edificios hace que la integración de los discapacitados en el mundo laboral se vea dificultada; lo que se pretende aquí es gracias al uso de la tecnología, desarrollar nuevos mecanismos de control y análisis que permitan ofrecer una solución a esta problemática, reduciendo en la medida que sea posible los problemas de accesibilidad a los que se vean sometidos los trabajadores.

En este caso, el esquema por el que se rigen los entornos laborales que se pretenden modelar es similar al de las organizaciones virtuales de agentes, ya que siguen una cierta estructura jerarquizada, tienden a la organización en diferentes grupos y cada elemento busca la consecución de ciertos objetivos para llevar a cabo la resolución de problemas más globales a través de la cooperación entre los elementos de la organización. Los objetivos de este trabajo pretenden contribuir en el problema propuesto inicialmente, la facilitación de la integración de personas discapacitadas en entornos laborales, con ese fin se plantean llevar a cabo los siguientes objetivos:

- Realización de un **análisis del estado del arte** en agentes y sistemas multi-agente: donde se estudiarán los conceptos básicos relativos al paradigma de

agentes, así como de los sistemas multi-agente, organizaciones virtuales de agentes, simulación social basada en agentes y diferentes metodologías y herramientas para su desarrollo.

- Estudio de diferentes **herramientas de representación 3D** con las que podamos dar soporte a la visualización del proceso que se desea simular.
- Realizar el **modelado de un entorno de oficina utilizando organizaciones virtuales de agentes**, de manera que podamos establecer diferentes grupos, roles, normas y tareas que nos permitan representar ese entorno de la manera más fiel posible a la realidad.
- Desarrollo de una **herramienta de visualización** que represente los elementos y procesos que se llevan a cabo en la organización virtual de agentes, implementando para ello los mecanismos que sean necesarios para la sincronización entre la plataforma de agentes y la herramienta de visualización.
- Investigar las diferentes **técnicas de simulación de agentes** que ayuden a **determinar los problemas de accesibilidad** existentes a partir del modelo de organizaciones de agentes, las interacciones posibles y la legislación actual.

3. Estado del arte

El estudio de los sistemas multi-agente surge en el campo de la inteligencia artificial distribuida [3] [4], aunando conceptos e ideas de varios campos, como la inteligencia artificial, ciencias computacionales, sociología, economía, ciencias organizacionales y la filosofía. Actualmente es un campo activo no sólo en el ámbito de la investigación y la enseñanza, sino además de aplicación industrial y comercial. Una de las primeras definiciones del término de agente la aporta [5] donde establece que *un agente es un sistema computacional encapsulado basado en hardware o (más habitualmente) basado en software, situado en algún entorno, que disfruta de las propiedades de autonomía, habilidades sociales, reactividad y proactividad y que actúa en ese entorno para poder alcanzar sus objetivos de diseño*. Para definir el concepto de agente de manera que se unifiquen las distintas definiciones existentes, se establece que un agente es una entidad que presenta una serie de características: autonomía, reactividad, iniciativa, sociabilidad, inteligencia, organización, movilidad, aprendizaje, veracidad y benevolencia.

Más adelante surge el concepto de Sistemas Multi-agente, que tratan sobre la coordinación inteligente de un conjunto de agentes autónomos, viendo cómo coordinar sus conocimientos, metas, propiedades y planes para la resolución de problemas o la toma de decisiones [6]. Según [7], un sistema multi-agente ideal tiene una serie de características: los entornos multi-agente proporcionan una infraestructura a través de protocolos de comunicación e interacción, son sistemas no centralizados y abiertos y los agentes que lo forman son autónomos, heterogéneos y distribuidos, pudiendo ser cooperativos o egocentristas. Siguiendo esta tendencia, será necesario definir conceptos de reglas [8], normas e instituciones [9] y estructuras sociales [10], que surgen de la necesidad de contar con un nivel

de abstracción mayor –independiente del agente–, que defina explícitamente la organización en la que los agentes conviven. En este contexto, surge el concepto de sociedad de agentes, introducido en [11] como el conjunto de entidades artificiales interrelacionadas e interactuantes, que se rigen bajo determinadas reglas y condiciones. Las arquitecturas empleadas para modelar y construir sistemas multi-agente basados en organizaciones, deben dar soporte a la coordinación entre agentes, de manera que se adapten dinámicamente a cambios en su estructura, objetivos o interacciones [12], permitiendo así la coexistencia de los agentes en entornos compartidos, de tal forma que puedan llevar a cabo sus objetivos. Cuando un agente forma parte de una sociedad de agentes, toma una visión colectivista, por lo que debe tener en cuenta el funcionamiento global del sistema, cómo unos agentes influyen en otros, de forma que no se centre en sí mismo.

El modelado basado en agentes es la técnica más apropiada para la simulación de sistemas con un gran número de objetos activos, bien sean personas, unidades de negocio, animales, ... que dependen del orden o tiempo en que los eventos ocurren, ya que ofrecen la capacidad de capturar entornos dinámicos de alta complejidad, pueden implementarse sin necesidad de tener un gran conocimiento acerca de las interdependencias globales y debido a la facilidad que ofrecen a la hora de construirlos frente a cambios en el modelo, ya que generalmente sólo es necesario realizar cambios locales, sino globales [13]. Por ello, el uso de técnicas basadas en agentes inteligentes es idóneo para desarrollar sistemas con los que se obtengan simulaciones de entornos cambiantes y complejos, surgiendo así el concepto de simulación social basada en agentes (ABSS, Agent-Based Social Simulation).

La simulación social basada en agentes es una disciplina que se ocupa de la simulación de fenómenos sociales, utilizando modelos computacionales multi-agentes. En estas simulaciones, se representa a las personas o grupos de personas como agentes. La simulación social basada en agentes es por tanto, una combinación de ciencias sociales (conjunto de ciencias que estudian la interacción entre entidades sociales), computación basada en agentes y simulación computacional [14], [15]. El fenómeno a simular es un conjunto de secuencias o eventos en un sistema (natural o artificial), que puede existir o no en el momento en que se realiza la simulación. Generalmente la simulación se lleva a cabo para obtener un mayor conocimiento del fenómeno que se estudia, pudiendo por ejemplo, probar diferentes modelos de sistemas, realizar predicción de comportamientos futuros o realizar experimentos de rendimiento que, de no realizarse mediante una simulación, no podrían llevarse a cabo.

4. Propuesta

Una de las metas de este trabajo, consiste en realizar el diseño y desarrollo de una herramienta que permita visualizar de manera tridimensional toda la información recopilada de un entorno laboral de oficina. Así, será posible investigar en nuevos mecanismos y técnicas que permitan visualizar el entorno simulado, así como tomar decisiones sobre la mejora de los procesos de trabajo que ocurran

en el entorno, con el objetivo fundamental de identificar las diferentes barreras arquitectónicas de éste, pudiendo mejorar de esta manera la accesibilidad del entorno de trabajo a personas con distintas discapacidades.

El uso de los sistemas multi-agente se adapta a la perfección en este contexto, debido a que un entorno de oficina se asocia con un sistema abierto (heterogéneo y dinámico). Además, permite la inclusión de técnicas que permitirán realizar una recopilación de información y una observación de los elementos del sistema de manera continuada y estructurada, determinando así cuáles son los elementos fundamentales de los trabajadores que se están analizando, así como los procedimientos de la empresa, los servicios disponibles en el lugar de trabajo y su configuración física.

El sistema propuesto permitirá simular un entorno laboral haciendo uso de la información disponible acerca de los procesos que ocurran en un entorno laboral, proponiendo tras la realización de las simulaciones una serie de mejoras al proceso de desarrollo del trabajo, gracias al análisis de los comportamientos y evoluciones del entorno de trabajo, que faciliten una integración para personas con discapacidad adecuada a sus necesidades.

Para llevar a cabo la simulación del proceso deseado, deberemos pasar por distintas etapas: (1) Definición del modelo del edificio en que se desee llevar a cabo la simulación a través de un editor 3D. (2) Definición de la estructura del sistema multi-agente que modele la organización de la empresa que se desea simular, estableciendo los distintos roles y tareas que intervengan en los procesos de la empresa. (3) Despliegue de una herramienta de visualización que permita ver cómo se desarrolla el proceso de simulación a través de un entorno tridimensional. (4) Extracción del conocimiento generado por la plataforma, que deberá determinar una serie de recomendaciones para solventar los problemas de accesibilidad a los que se han visto sometidos los agentes que han intervenido en la simulación.

4.1. Modelo arquitectónico de simulación

En la figura 1 podemos ver la arquitectura general del sistema, que estará conformado por tres grandes bloques: un editor 3D, un sistema multi-agente y un visualizador 3D.

En el editor 3D podremos realizar el modelado de los edificios, delimitando las habitaciones y los puestos de trabajo. El uso de este editor hará posible definir las paredes, suelos, puertas, localización de las ventanas, texturas para los suelos y paredes, escaleras u objetos dentro de las habitaciones (sillas, mesas, plantas, y una gran variedad de elementos). Los modelos de los distintos edificios pueden ser codificados en un formato específico, con el propósito de almacenar remotamente la definición del edificio en una plataforma *Cloud Computing* de la que se importarán y exportarán los escenarios que formen parte de las simulaciones.

El sistema multi-agente, estará compuesto de una serie de agentes inteligentes que permiten realizar la simulación del entorno de oficina. Para el modelado de éste, se seguirá el enfoque de las metodologías orientadas a la organización,

que tratan de describir de manera general los objetivos de la organización, su estructura organizativa, dinámica y entorno de la organización.

Se definirán agentes de los diferentes departamentos que haya en la empresa, cada uno de los cuales realiza unas actividades específicas para dicho departamento, además de las actividades comunes para cualquier agente (persona física) de cualquier departamento. Por otro lado, los agentes se modelan con diferentes tipos de discapacidades, por lo que la forma de realizar una misma actividad variará para un agente con una discapacidad a otro o no discapacitado o que tenga una discapacidad diferente. La estructura de la organización de agentes se almacenará en una base de datos que nos permita gestionar los agentes existentes dentro del sistema multi-agente, pudiendo definir qué agentes están presentes en éste, así como los roles que tienen asignados, y poder modelar la estructura de la organización virtual,

El visualizador 3D nos dará el soporte necesario para representar y renderizar el edificio y las características definidas previamente en el editor 3D. También se encarga de mostrar los diferentes agentes que forman parte del sistema, haciendo posible visualizar el desempeño de las diferentes actividades de cada agente.

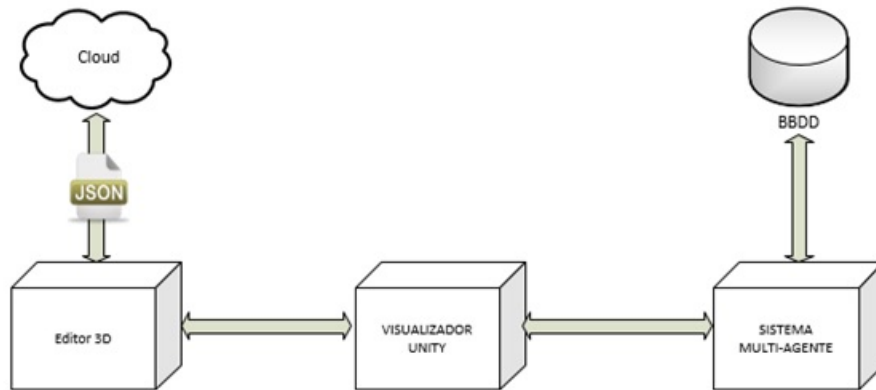


Figura 1. Arquitectura del sistema.

4.2. Arquitectura de agentes

Los agentes dentro de un sistema multi-agente tratan, de manera cooperativa, de llevar a cabo tareas dentro del entorno del sistema al que pertenecen. Las tareas que se puedan realizar dentro del sistema vendrán determinadas por el conocimiento y habilidades de los agentes y de las restricciones que imponga el entorno.

En este contexto, los agentes representarán los distintos elementos y procesos que se desarrollan en el ámbito laboral, pudiendo modelar gracias a ellos las diferentes tareas atendiendo al rol que éstos tengan, así como de las posibles

discapacidades que puedan tener, que influirán en la forma en que lleven a cabo dichas tareas. Además, el entorno puede presentar diferentes barreras arquitectónicas que impidan o dificulten la ejecución de las tareas, principalmente en los agentes que representen a un trabajador discapacitado. El objetivo final del sistema es conseguir minimizar las tareas que no se puedan realizar para así promover la integración de todos los agentes dentro del entorno. Para ello, gracias a la simulación de diferentes tareas seremos capaces de detectar aquellas que sean especialmente críticas, pudiendo después utilizar distintas configuraciones del entorno o de la planificación de tareas para tratar de hacer que el entorno sea accesible.

Uno de los objetivos que se buscó a la hora de desarrollar esta herramienta fue el de conseguir modelar de manera flexible distintos entornos laborales, por ello, la estructura del sistema multi-agente en cada simulación podrá variar, generando componentes específicos en función del entorno a modelar.

De manera genérica, podemos determinar que una compañía está dividida en diferentes áreas (dirección, ventas, recursos humanos, administración, logística, etc.), con uno o varios responsables de área que planifican el trabajo de los empleados que estén a su cargo. Ante esta situación, debemos ser capaces de establecer de manera análoga y genérica las distintas áreas funcionales que conforman la empresa, por lo que se ha optado por modelar el sistema multi-agente como una organización virtual de agentes con una estructura jerárquica, donde en los niveles superiores se situarán los agentes encargados de la toma de decisiones -determinando qué tareas es necesario realizar y quién debe realizarlas- y en las capas más bajas los agentes que tengan una funcionalidad más básica -encargados de llevar a cabo las tareas designadas-.

Tras ver la dimensión estructural general que se plantea modelar, es necesario definir qué elementos existirán en el sistema, ver cómo se relacionan entre sí qué funcionalidad específica tendrán (en base a sus servicios, tareas y objetivos, la dimensión y contexto del entorno del sistema y las distintas normas por las que se rija la organización. Para realizar el modelado del sistema, se seguirá la metodología GORMAS, por lo que primero es necesario establecer el modelo de interacción, analizar las necesidades expectativas de los usuarios que utilicen el sistema y la forma en que se realiza el intercambio de información. Definimos los siguientes roles:

- **Usuario:** representa al usuario o cliente del sistema, en este contexto podrá ser el responsable de definir la distribución de elementos en el edificio, el responsable de recursos humanos que asigne los puestos de trabajo a los empleados o el arquitecto encargado de diseñar el edificio en que se llevará a cabo la actividad profesional, con el fin de obtener información acerca de la accesibilidad del entorno.
 - Es el encargado de iniciar el proceso de simulación.
 - Modelará los distintos agentes que hagan de actores en representación de los trabajadores de la empresa, así como las tareas que éstos vayan a realizar durante la simulación.

- Tiene acceso a la información que se genera tras los procesos que se simulan.
- **Dirección:** este agente se encarga de realizar la planificación de tareas que se llevará a cabo por el conjunto de la organización. Podrá estar presente o no dependiendo de la estructura organizacional de la empresa a representar.
 - Genera las tareas que representen los objetivos a cumplir por el conjunto de la organización.
 - Delega su distribución a los diferentes responsables de área atendiendo a la naturaleza de la tarea.
- **Responsable de área/departamento:** planifica la distribución de las tareas asignadas al departamento que gestiona.
 - Recibe las tareas de dirección
 - Planifica su distribución entre los agentes que forman el departamento atendiendo a su disponibilidad (tendrán mayor prioridad los agentes que tengan asignados un menor número de tareas) y capacidades (no todos los agentes ofrecen los mismos servicios).
- **Trabajador:** agentes que representan cada uno de los trabajadores que intervienen en la simulación.
 - Los agentes tendrán una pila donde almacenen todas las tareas que deben realizar, previamente asignadas a través del responsable de área.
 - Emitirán información relativa al grado de éxito con que han llevado a cabo cada una de ellas.
 - Todos estos agentes tienen en común una serie de elementos, que definen un conjunto de comportamientos y capacidades cognitivas compartidas entre ellos. Además de estas características comunes, los agentes tendrán características propias, definidas en función de su rol dentro de la organización y su discapacidad.
 - Los roles asignados determinarán las tareas que puedan realizar y comportamientos específicos para cada una de ellas, de manera que puedan modelarse ejecuciones diferentes para una misma tarea atendiendo al tipo de discapacidad del trabajador.
- **Agentes de entorno:** representan aquellos elementos del edificio con los que el agente puede interactuar de manera directa. Podemos encontrar ejemplos dentro de este tipo de agentes en elementos del entorno como pueden ser los teléfonos, ascensores o fotocopiadoras.
 - Estos agentes no son conscientes del entorno que les rodea, sólo son conscientes de su estado interno.
 - Cualquier interacción con otros agentes será iniciada en cualquier caso por un agente humano.
 - Cuando un agente solicite su uso, éste consultará su estado interno, comprobando si está libre y además tiene los recursos necesarios para que se pueda llevar a cabo la tarea solicitada, y comunicará esta información al agente, determinando así si puede finalizar la tarea que se le había asignado.

- **Agente Base de Datos:** este agente ofrece una serie de servicios que facilitan todas las formas de acceso que se tienen sobre la base de datos que almacenará la estructura de la Organización Virtual.
 - Cuidará de accesos inadecuados por parte de agentes que no debieran poder consultar o modificar el contenido de la base de datos. Estos servicios permitirán añadir agentes, roles, servicios, etc. a la base de datos, así como consultar información, modificar, borrar su información.
 - Para el acceso a estos servicios será necesario un pequeño módulo para comprobar permisos, haciendo los servicios accesibles únicamente a los agentes que se especifique.
- **Agente de recursos humanos:** La organización virtual propuesta puede clasificarse dentro de las llamadas sociedades semi-abiertas, ya que tendrá un mecanismo que controle la admisión de los agentes, siendo necesario realizar una petición al mecanismo para que este evalúe la entrada o no del agente a la sociedad. Este agente es el encargado de llevar a cabo dicha tarea.
 - Evaluará las candidaturas de los agentes que soliciten acceder a la organización, aprobando las distintas propuestas en función de las plazas vacantes que existan en el departamento al que intente acceder el agente.
 - El número de plazas por departamento vendrá definido por los puestos de trabajo que se definan en el entorno laboral, que tendrán asociado un departamento y un agente en caso de estar ocupados.
- **Agente comunicación:** encargado de comunicar el sistema multi-agente con la herramienta de visualización, de forma que facilite el paso de mensajes entre los dos módulos de la arquitectura.
 - Garantiza que exista una correspondencia real entre lo que ocurre en la plataforma de agentes y la visualización que tendrá lugar en el visualizador.
- **Agente supervisor:** analizará los distintos comportamientos y evoluciones de los agentes que intervienen en la simulación para así poder llevar a cabo un análisis de los procesos que tienen lugar en la organización.
 - Se comunicará con el resto de agentes tras la consecución de una tarea, recogiendo si han podido llevarla a cabo o no.
 - En el caso de que una tarea se haya realizado satisfactoriamente se comunicará qué tarea ha sido realizada por qué agente, indicando el tiempo y recursos que han sido necesarios para realizarla. En caso contrario, el agente implicado indicará al agente supervisor los motivos por los cuales ésta no se haya podido realizar.

Atendiendo a la metodología utilizada, se deberá instanciar la vista funcional del modelo de organización, que representa los productos y servicios que ofrece el sistema, el tipo de entorno, los objetivos globales que persigue, los grupos de interés a los que afecta y la información que consume. Se muestra el diagrama de dicho modelo en la figura 2.

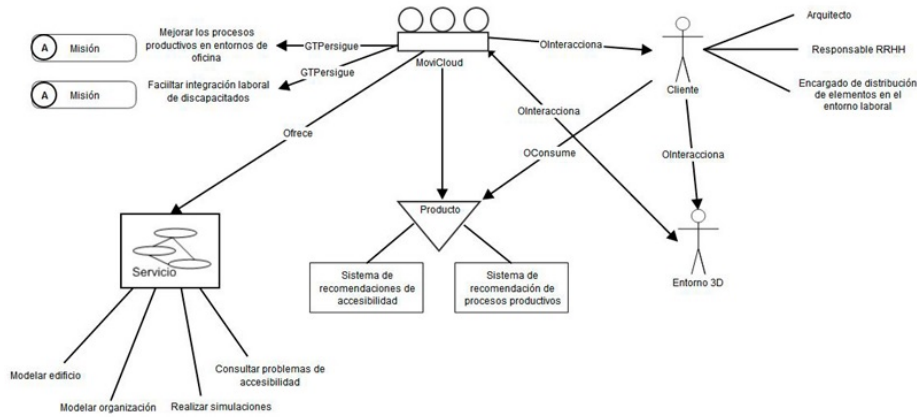


Figura 2. Diagrama del modelo de organizaci3n (vista funcional). Misi3n del sistema.

4.3. Proceso de asignaci3n de tareas

En la propuesta realizada, para poder modelar una tarea en t3rminos de agentes, es necesario en primer lugar la existencia de un agente que represente al empleado que va a realizar la tarea y otro agente que solicite la realizaci3n de la tarea. La solicitud de una tarea puede darse de dos formas diferentes: bien que el usuario indique expl3citamente qu3 tarea debe realizar un agente en concreto, o bien que un agente dentro de la organizaci3n, tras planificar el reparto de tareas, solicite a otro que realice la tarea. En el primer caso, para llevar a cabo la solicitud de la tarea por parte del usuario, se crear3 un agente temporal que inicie la comunicaci3n con el agente al que se le designe la tarea, que tras la conclusi3n de esta, ser3 eliminado de la organizaci3n. En el segundo caso no ser3 necesaria la creaci3n de nuevos agentes para la solicitud de tareas, ya que por cada departamento existir3 necesariamente un agente responsable que se encargar3 de ello.

Para llevar a cabo la comunicaci3n entre los agentes se utilizar3n mensajes FIPA-ACL, siguiendo el protocolo FIPA-Request Interaction Protocol, el cual permite que el agente que inicia la conversaci3n realice una petici3n a otro participante para que lleve a cabo alguna acci3n, por lo que se ajusta a la perfecci3n a la comunicaci3n que deben seguir los dos roles que intervendr3n en la conversaci3n. En la figura 3 podemos ver un ejemplo de la aplicaci3n de este protocolo a un ejemplo pr3ctico que se puede dar en nuestra organizaci3n, donde un responsable de 3rea solicita a un trabajador a su cargo la realizaci3n de una tarea simple, hacer unas fotocopias. En el flujo que sigue este protocolo primero el responsable iniciar3 la conversaci3n con un mensaje de tipo *request*, con el que solicita la realizaci3n de la tarea por parte del trabajador; 3ste puede responder con tres tipos de mensaje distintos: *agree*, con el que indica que pasar3 a realizar las fotocopias, *notUnderstood*, indicando en este caso que no ha sido capaz de

comprender el mensaje con el que se le ha solicitado la realización de la tarea, o *refuse*, con el que comunica que no ha podido realizar las fotocopias (puede que el agente esté ocupado). Los actos comunicativos *notUnderstood* y *refuse* suponen que la conversación sea finalizada, mientras que en el caso de *agree* el trabajador mandará un último mensaje al iniciador de la conversación, que puede ser de tipo *inform*, en el caso de que la tarea (realizar las fotocopias) se haya realizado satisfactoriamente o *failure* si no ha sido así por algún motivo (la fotocopidora puede estar estropeada, ésta no es accesible al agente, etc.).

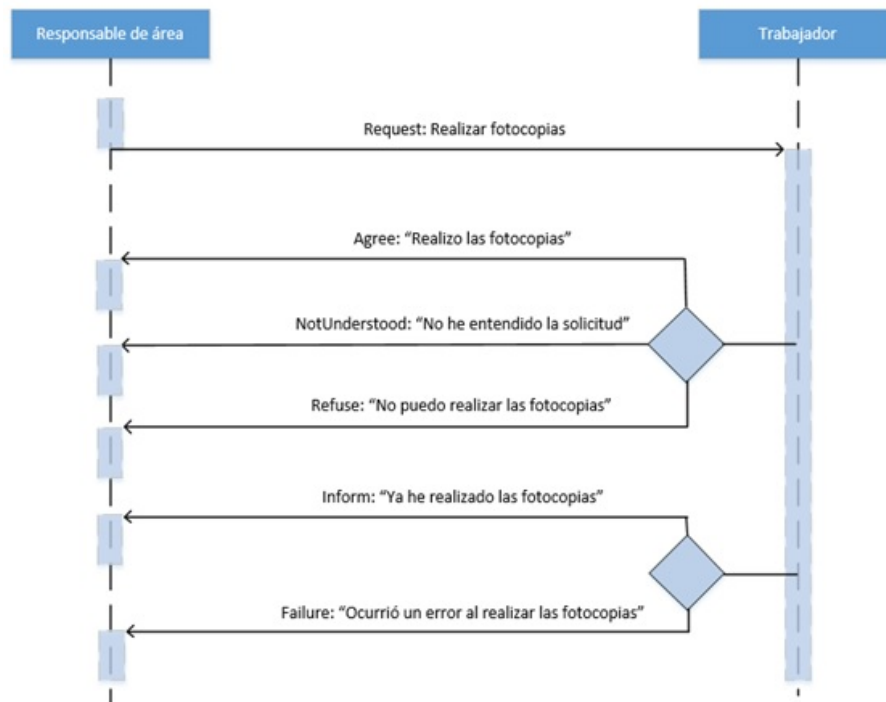


Figura 3. Ejemplo FIPA-Request Interaction Protocol entre dos trabajadores

Por norma general, esta es la lógica que se seguirá en el trabajo. Además es importante que exista una sincronización de las tareas con el visualizador 3d, de manera que exista una correspondencia directa entre lo que ocurre en la plataforma de agentes con lo que se está visualizando

4.4. Gestión de la organización virtual

Como hemos introducido, JADE no da soporte a la gestión de organizaciones virtuales de agentes, por lo que ha sido necesario introducir una funcionalidad

adicional con la que poder realizar esta tarea. Para la gestión de los agentes existentes en la organización virtual, definiendo qué agentes están presentes en la misma, a qué departamento pertenecen, los roles que tienen asignados, o los servicios que ofrecen, así como características adicionales que puedan presentar los agentes, se utilizará una base de datos.

Ya definimos previamente que existirá un agente encargado de la gestión de esta base de datos, el **Agente Base de Datos**. Este agente ofrece una serie de servicios que facilitan todas las formas de acceso que se tienen sobre la base de datos que almacenará la estructura de la Organización Virtual, cuidando de accesos inadecuados por parte de agentes que no debieran poder consultar o modificar el contenido de la base de datos. Estos servicios permitirán añadir agentes, roles, servicios, etc. a la base de datos, así como consultar información, modificar, borrar su información. Para el acceso a estos servicios se implementa un pequeño módulo para comprobar permisos, haciendo los servicios accesibles únicamente a los agentes que se especifique. Por ejemplo, un agente que no sea de Recursos Humanos no tiene permisos para añadir agentes a la Organización Virtual. Para ello, el Agente Base de Datos recibirá una instancia del agente que realiza la consulta, gracias a la cual comprobará qué rol tiene el agente y qué tipo de permisos tiene sobre la base de datos. En la figura 4 se muestra la estructura de la base de datos que se utilizará para almacenar la estructura de la organización virtual:



Figura 4. Estructura de la BBDD para modelar la OV.

4.5. Comunicación de la plataforma de agentes con la herramienta de visualización

Una de las tareas esenciales dentro de la plataforma será la de establecer una comunicación entre JADE y el entorno 3D (implementado en Unity 3D) para que exista una correspondencia real entre lo que ocurre en la plataforma de agentes y la visualización. Dentro de la plataforma de agentes, existirá un agente encargado de gestionar las comunicaciones en la parte de JADE.

Con este propósito, la comunicación entre la plataforma de agentes y el entorno 3D se realizará mediante sockets TCP, existiendo en Unity un módulo dedicado a la gestión de las peticiones realizadas la plataforma de agentes y viceversa. Las tareas de creación, eliminación e interacción entre agentes ejecutadas en la plataforma deben ser actualizadas en el entorno 3D, y las tareas de creación, eliminación e interacción entre agentes ejecutadas desde la interfaz 3D, deben llevarse a cabo primero en la plataforma de agentes para pasar a ser posteriormente actualizadas en Unity.

La comunicación se dividirá en 2 bloques:

- Sockets dedicados a la realización de las tareas en JADE y su actualización correspondiente en Unity.
- Sockets dedicados a la realización de las tareas inicializadas desde la interfaz de Unity, que deben realizarse primero en JADE para después actualizarse en Unity.

En cuanto a la realización de tareas en JADE que serán actualizadas después en Unity, se utilizarán tres elementos: dos sockets, uno para el envío de tareas y otro para la confirmación de las tareas realizadas, y una tabla con la que mantendremos un registro de estas tareas.

En lo relativo a la realización de tareas en Unity y actualizadas en JADE, la tabla de tareas reside en el lado de Unity, ya que es el encargado de realizar las peticiones y debe comprobar que todo lo que se realice en JADE se actualice después en Unity.

La petición que se puede realizar desde Unity puede ser de tres tipos: creación de agentes., eliminación de agentes o asignación de tareas a los agentes.

El **Agente Comunicación** será el encargado de atender las peticiones que llegan desde Unity, realizando después las tareas que correspondan en función de la trama intercambiada.

4.6. Entorno 3D

Una organización está rodeada de un entorno que se encuentra en continuo cambio. En este caso el modelo del entorno vendrá determinado por la fase de diseño del edificio en que se desarrolle la simulación. Este proceso se realizará a través de una herramienta basada en Unity 3D, que nos permitirá generar entornos virtuales en 3D en escala, pudiendo definir elementos como plantas, paredes, pisos o suelos, además de incluir una serie de modelos de mobiliario. Este editor utiliza una serie de servicios de una plataforma de Cloud Computing

donde que se almacenarán y recuperarán los modelos de los edificios que hayamos diseñado.

Para la adaptación de esta herramienta al sistema que se desea desarrollar, ha sido necesario introducir una funcionalidad adicional que permita definir los distintos puestos de trabajo disponibles en el edificio así como la delimitación de qué espacios dentro del edificio pertenecen a qué departamentos.

Es necesario que los agentes tengan la capacidad de recabar información acerca del entorno, de forma que puedan decidir cómo ejecutar las acciones que intenta realizar con el fin de satisfacer su objetivo. Aunque el módulo comunicación y de toma de decisiones del agente esté implementado en JADE, el módulo encargado de recabar la información del entorno deberá estar implementado en Unity, teniendo así acceso a la información de dónde se encuentran los distintos elementos del entorno.

Uno de los principales problemas al que nos enfrentamos en las escenas 3D es cuando un modelo necesita desplazarse de un punto a otro, por lo que es necesaria la utilización de un algoritmo que calcule la ruta. En este caso haremos uso de CritterAI, un sistema de navegación que ofrece funcionalidad relativa a generación de mallas de navegación, búsqueda de caminos y administración de movimientos. En cuanto al algoritmo de cálculo de rutas utilizado por esta herramienta, ofrece implementaciones de los algoritmos de búsqueda de Dijkstra y A*, que encontrarán el camino de menor coste entre un nodo origen y uno de destino. Los nodos sobre los que estos algoritmos buscarán el camino de coste mínimo serán los generados en la malla de navegación.

Una vez definido el modelo del edificio, podremos pasar a realizar la simulación de una organización de agentes dentro de éste, para ello deberemos desplegar la herramienta de visualización, que al iniciarse deberá realizar una serie de operaciones:

- Cargar el modelo del edificio
- Generar la malla de navegación
- Sincronización con la plataforma de agentes

Esta herramienta dispondrá de una interfaz gráfica con la que podremos interactuar y modificar la organización virtual de agentes. Las operaciones que podemos llevar a cabo serán la de creación y eliminación de agentes y asignación de tareas. Como se ha tratado en el apartado anterior, todas estas operaciones solicitadas desde el entorno 3D se comunicarán a la plataforma de agentes, que atenderá estas solicitudes y confirmará o no su realización para poder así actualizar en consecuencia el estado de la visualización.

5. Conclusiones

En este trabajo se ha llevado a cabo una investigación relativa al desarrollo de un sistema multi-agente basado en organizaciones virtuales de agentes en el ámbito de la integración laboral de personas con discapacidades. El resultado final es una herramienta que facilita a las personas con discapacidad lograr una

igualdad de oportunidades (con ausencia total de discriminación) en el ámbito laboral, mediante el empleo de medios tecnológicos que consigan alcanzar la accesibilidad universal en los entornos donde desarrollen su actividad.

Para ello ha sido necesario realizar un análisis del estado del arte relativo a agentes, sistemas multi-agentes y organizaciones virtuales de agentes de manera genérica para después centrarse en el ámbito de aplicación de estas tecnologías para este caso de estudio: la simulación social basada en agentes, donde se han revisado distintas investigaciones y aplicaciones en casos prácticos para tener una referencia de su uso en distintos ámbitos.

Adicionalmente se han estudiado las distintas metodologías, protocolos y tecnologías existentes para poder implementar sistemas multi-agente de manera que se consiga con su uso la simplificación del desarrollo de este tipo de sistemas.

El núcleo del trabajo realizado se centra en la creación de una organización virtual de agentes con la que se consiga modelar de manera flexible distintos entornos laborales. Se ha optado por modelar su estructura de una manera flexible para así permitir su adaptación a distintas configuraciones de la estructura laboral de forma que podamos representar distintas tipologías de empresas.

Para concluir, podemos exponer que se han llevado a cabo de manera satisfactoria las investigaciones del estado del arte previstos en los objetivos que se marcaron inicialmente en lo relativo a agentes inteligentes, particularmente en el ámbito de la simulación social basada en agentes y de las diferentes opciones a través de las cuales podemos representar estos sistemas en entornos tridimensionales. Finalmente, se ha logrado tanto el diseño como la implementación de la herramienta propuesta, con la que modelar una organización virtual de agentes en representación de una organización del ámbito laboral para simular los procesos que en ésta se lleven a cabo, con el objeto de determinar los problemas de accesibilidad que puedan ocurrir en el entorno laboral en que se desempeñe la actividad laboral de los trabajadores representados.

5.1. Líneas futuras de investigación

En este apartado se muestran las distintas líneas de investigación y trabajo futuro que se han planteado en el contexto de este trabajo:

- Posibilidad de realizar una abstracción del modelo del edificio a través de ontologías, de manera que la información del entorno no resida únicamente en el entorno tridimensional, sino que la plataforma de agentes también tenga conocimiento de éste. Su integración con el sistema sería sencilla gracias a que JADE incorpora una serie de clases para el manejo de ontologías. El modelado de la ontología podría realizarse mediante OWL para facilitar la sintaxis, y exportado a JADE utilizando la herramienta Protégé.
- Para aprovechar las herramientas desarrolladas, permitir el modelado de otro tipo de organizaciones, no sólo del ámbito laboral, dando como lugar una herramienta más genérica que pueda ser utilizada para otros campos de estudio de la simulación social basada en agentes.

- Incorporación de alertas automáticas en el editor de edificios, que indiquen en tiempo de ejecución aquellas normativas de accesibilidad que se están incumpliendo.
- Permitir el desarrollo de diferentes entornos tridimensionales, no viéndonos limitados únicamente a la representación en edificios.
- Añadir funcionalidad adicional a la visualización, de forma que existan alternativas a la hora de extraer información sobre los procesos, como por ejemplo grafos que muestren de qué forma se dan las relaciones entre aquellos agentes que interactúan entre ellos o la generación de mapas de calor con los que quede constancia de las rutas tomadas por cada trabajador en el entorno tridimensional.
- Mejorar la extracción de conocimiento mejorando los algoritmos que intervienen en las recomendaciones de accesibilidad, ofreciendo además un posible servicio de estadísticas en tiempo real y un servicio de generación de informes.

Bibliografía

- [1] “Onu factsheet on persons with disabilities.” <http://www.un.org/disabilities/default.asp?id=18>.
- [2] “Normas uniformes sobre la igualdad de oportunidades para las personas con discapacidad,” 1994.
- [3] C. Roda, “Software agents, edited by jeffrey m. bradshaw. aaai press/the mit press, 1997, £31.95, isbn 0-262-52234-9,” *Knowl. Eng. Rev.*, vol. 14, no. 2, pp. 181–185, 1999.
- [4] J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1999.
- [5] M. Wooldridge and N. R. Jennings, “Intelligent agents: theory and practice,” *The Knowledge Engineering Review*, vol. 10, pp. 115–152, 6 1995.
- [6] J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1999.
- [7] M. N. Huhns and L. M. Stephens, “Multiagent systems,” ch. Multiagent Systems and Societies of Agents, pp. 79–120, Cambridge, MA, USA: MIT Press, 1999.
- [8] F. Zambonelli, “Abstractions and infrastructures for the design and development of mobile agent organizations,” in *Agent-Oriented Software Engineering II* (M. Wooldridge, G. Weiß, and P. Ciancarini, eds.), vol. 2222 of *Lecture Notes in Computer Science*, pp. 245–262, Springer Berlin Heidelberg, 2002.
- [9] M. Esteva, J.-A. Rodríguez-Aguilar, C. Sierra, P. Garcia, and J. Arcos, “On the formal specification of electronic institutions,” in *Agent Mediated Electronic Commerce* (F. Dignum and C. Sierra, eds.), vol. 1991 of *Lecture Notes in Computer Science*, pp. 126–147, Springer Berlin Heidelberg, 2001.
- [10] H. Van Dyke Parunak and J. Odell, “Representing social structures in uml,” in *Agent-Oriented Software Engineering II* (M. Wooldridge, G. Weiß, and P. Ciancarini, eds.), vol. 2222 of *Lecture Notes in Computer Science*, pp. 1–16, Springer Berlin Heidelberg, 2002.
- [11] M. Annunziato and P. Pierucci, “The emergence of social learning in artificial societies,” in *Applications of Evolutionary Computing* (S. Cagnoni, C. Johnson, J. Cardalda, E. Marchiori, D. Corne, J.-A. Meyer, J. Gottlieb, M. Middendorf, A. Guillot, G. Raidl, and E. Hart, eds.), vol. 2611 of *Lecture Notes in Computer Science*, pp. 467–478, Springer Berlin Heidelberg, 2003.
- [12] V. Dignum, *A model for organizational interaction: based on agents, founded in logic*. PhD thesis, Universiteit Utrecht, 2004.
- [13] A. Borshchev and A. Filippov, “From system dynamics and discrete event to practical agent based modeling: reasons, techniques, tools,” in *In Proceedings of the 22nd international conference of the system dynamics society*, vol. 22, 2004.

- [14] X. Li, W. Mao, D. Zeng, and F.-Y. Wang, “Agent-based social simulation and modeling in social computing,” in *Intelligence and Security Informatics* (C. Yang, H. Chen, M. Chau, K. Chang, S.-D. Lang, P. Chen, R. Hsieh, D. Zeng, F.-Y. Wang, K. Carley, W. Mao, and J. Zhan, eds.), vol. 5075 of *Lecture Notes in Computer Science*, pp. 401–412, Springer Berlin Heidelberg, 2008.
- [15] P. Davidsson, “Agent based social simulation: A computer science view,” *Journal of artificial societies and social simulation*, vol. 5, no. 1, 2002.

Plataforma multiagente adaptable al contexto sobre sistema *cloud*

Víctor Parra Vidales¹, Juan Francisco de Paz Santana¹ y Belén Pérez Lancho¹

¹Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España
`{parra,fcfods,lanchos}@usal.es`

Resumen El objetivo principal de este trabajo busca diseñar un sistema multiagente basado en organizaciones virtuales que sea capaz de gestionar el entorno de un usuario, adaptándose a sus necesidades. Además, gracias a la recolección de los datos que proporcionen los sensores instalados en la vivienda y en los dispositivos móviles del usuario, se pretende obtener un banco de datos que, tras el uso continuado del sistema, pueda proporcionar información acerca de las rutinas del usuario, con el fin de automatizar tareas y ayudar en la gestión energética de la vivienda, en las labores de vigilancia y seguridad y en el confort de los habitantes.

Con la instalación de un prototipo se pretende iniciar la recolección de datos referentes al ambiente de una vivienda durante un periodo no inferior a un año, con el fin de tratarlos y observar hasta qué punto el sistema puede ser viable en un entorno real.

Para el desarrollo del trabajo se han estudiado los sistemas multiagente, los esquemas de organización virtual. Además se han planteado distintos métodos de generación de reglas, tanto de manera directa por el usuario como de manera automática gracias al uso de clasificadores basados en datos históricos. En una primera aproximación, planteada en este trabajo, se hace uso de árboles de decisión. Estos árboles se generan a partir de la obtención de datos simulados, con el fin de recrear las acciones ejecutadas por el sistema o el usuario durante un día y una vez el sistema ha sido debidamente entrenado con el conjunto de datos mínimamente necesarios.

Finalmente, se exponen las conclusiones extraídas tras la investigación sobre los distintos métodos de tratamiento de datos, la utilidad de estos métodos en el mundo actual y la aplicabilidad de los mismos a la cantidad de información que hoy en día se registra de cada persona de manera individual.

Keywords: Sistemas multiagente, Arquitectura distribuida, Árboles de decisión

1. Introducción

La planificación y automatización de tareas en un contexto adaptable es uno de los mayores problemas que actualmente acaparan las líneas investigadoras en

el campo de [1, 2, 3, 4, 5]. El objetivo principal de estas investigaciones persigue siempre adaptar el funcionamiento de los sistemas al contexto en el que están inmersos, buscando lograr un comportamiento diferente en función de los eventos externos que puedan influir sobre el resultado.

En la actualidad, la integración de la vida diaria con la vida digital ha producido que muchas de nuestras acciones queden registradas gracias a los numerosos sensores que día a día monitorizan nuestros movimientos, sin ni siquiera darnos cuenta de ello a veces. Así, el simple hecho de llevar un *Smartphone* encima todo el día permite que se pueda extraer la información de por qué lugares hemos pasado, dónde solemos ir más, cuánto tiempo pasamos en cada sitio, cómo nos solemos desplazar, etc. Con esa información, las compañías pueden proporcionarnos información extra que pueda resultar de interés y publicidad personalizada en función de las zonas por las que nos desenvolvamos. Por otro lado, el uso de Internet para realizar cualquier búsqueda, compra, conversación... permite que una plataforma nos pueda sugerir actividades de nuestro interés, por ejemplo ciertos días del mes o de la semana, qué cenar y en qué restaurantes.

Por otra parte, los agentes y sistemas multiagentes (MAS) han ganado importancia en el desarrollo de aplicaciones en entornos dinámicos y flexibles, tales como la Web [6], sistemas de control [7], sistemas de recomendación o sistemas de simulación. Los agentes se caracterizan por sus capacidades, tales como autonomía, reactividad, pro-actividad, habilidades sociales, razonamiento, aprendizaje y movilidad entre otras. Las capacidades se pueden modelar de distintas formas y con diferentes metodologías [33].

Este trabajo presenta una arquitectura basada en la utilización de organizaciones virtuales de agentes que permite simular el funcionamiento de una vivienda de manera automatizada basándose en órdenes transmitidas por el usuario de manera directa a la plataforma y en la extracción de características que produce el usuario a la hora de interactuar de manera pasiva con el sistema. La planificación del sistema incluye tanto los recursos necesarios para la recolección de datos como para la distribución de las tareas y la asignación de funciones a cada uno de los agentes disponibles en el sistema.

1.1. Motivación y objetivos

El objetivo principal que persigue la investigación planteada en este trabajo es el de diseñar un sistema basado en organizaciones virtuales que permita la gestión automatizada de los recursos software y hardware disponibles en el entorno de un usuario, que lleve un registro de la actividad del usuario y pueda utilizarlo para adaptar el entorno del usuario a sus necesidades en función de la hora, día, mes y agenda de que dispone. Además debe disponer la información en un soporte *cloud* con el fin de que esté accesible desde cualquier agente que lo precise, sin dependencia de su ubicación física.

Respecto al objetivo final que se pretende alcanzar con este proyecto a largo plazo es desarrollar un sistema inteligente que sea capaz de adaptarse de manera continua al día a día de un usuario y gestione de manera eficiente la vivienda, con el fin de automatizar tareas que hasta ahora pueden repercutir en pérdidas de

tiempo innecesarias para el usuario, ahorrar costes en los gastos de energía optimizando el consumo. . . y cualquier función personalizada que pueda programar el usuario de manera directa para su ejecución automatizada.

Debido al alcance que puede llegar a tener el sistema, se pretende realizar una implantación de manera escalonada, con el fin de controlar las posibles inestabilidades que pudieran producirse. Inicialmente se empezará con una arquitectura simple que se centre en la gestión de la seguridad de la vivienda (controlando los accesos y la ubicación del usuario) junto con la monitorización de la temperatura con el fin de ahorrar costes en la climatización de la vivienda en función de la época del año que se corresponda en cada momento. Puesto que actualmente los actuadores sobre electrodomésticos suelen regirse por el control del paso de electricidad, no se tendrán en cuenta a la hora de su gestión real, sino que, de considerarse, serán simulados.

2. Estado del arte

La cantidad de datos que se llegan a generar hoy en día hace necesario que se utilicen nuevas técnicas de tratamiento de la información [9][10][11]. El uso de técnicas de minería y de extracción de características ha sido posible en el campo de la informática gracias a la evolución de los soportes hardware. Con ellos, el tratamiento de cantidades ingentes de datos ha permitido descubrir comportamientos y patrones en numerosos ámbitos de la vida [12]. En la actualidad, muchas empresas aplican estas técnicas para extraer modelos de mercado y lograr descubrir nuevas formas de atraer el dinero de los compradores.

Para hacernos una idea del volumen de datos que se llega a manejar en estos casos, nos basaremos en la información que proporciona IBM [13]: en los últimos meses se está generando más de 2.5 quintillones de bytes al día. Tal es el volumen de datos que *el 90 % de los datos del mundo han sido creados durante los últimos dos años*.

Por ello, la necesidad de disponer de un sistema *cloud* que soporte una gran cantidad de datos, y el poder disponer de ellos de manera remota, son dos factores que influyen a la hora de implantar un sistema que tenga como base el contexto y la ubicuidad, como es el caso que se presenta.

Además del gran volumen de información, ésta existe en una gran variedad de datos que pueden ser representados de diversas maneras en todo el mundo, por ejemplo de dispositivos móviles, audio, vídeo, sistemas GPS, incontables sensores digitales en equipos industriales, automóviles, medidores eléctricos, veletas, anemómetros, etc., los cuales pueden medir y comunicar el posicionamiento, movimiento, vibración, temperatura, humedad y hasta los cambios químicos que sufre el aire, de tal forma que las aplicaciones que analizan estos datos requieren que la velocidad de respuesta sea lo suficientemente rápida para lograr obtener la información correcta en el momento preciso. Estas son las características principales de una oportunidad para lo que se conoce como *Big Data*.

De acuerdo con un estudio realizado por Cisco [14], entre el 2011 y el 2016 la cantidad de tráfico de datos móviles crecerá a una tasa anual de 78 %, así

como el número de dispositivos móviles conectados a Internet excederá el número de habitantes en el planeta. Las Naciones Unidas proyectan que la población mundial alcanzará los 7.5 billones para el 2016 de tal modo que habrá cerca de 18.9 billones de dispositivos conectados a la red a escala mundial; esto conllevaría a que el tráfico global de datos móviles alcance 10.8 Exabytes mensuales o 130 Exabytes anuales. Este volumen de tráfico previsto para 2016 equivale a 33 billones de DVDs anuales o 813 cuatrillones de mensajes de texto.

Pero no solamente somos los seres humanos quienes contribuimos a este crecimiento enorme de información, existe también la comunicación denominada máquina a máquina (M2M, *machine-to-machine*) cuyo valor en la creación de grandes cantidades de datos también es muy importante. Por ejemplo es el caso de los sensores digitales instalados en contenedores para determinar la ruta generada durante una entrega de algún paquete y que esta información sea enviada a las compañías de transportes, o de sensores en medidores eléctricos para determinar el consumo de energía a intervalos regulares que envían esta información a las compañías del sector energético. Se estima que hay más de 30 millones de sensores interconectados en distintos sectores como automoción, transporte, industria, servicios, comercial, etc. y se espera que este número crezca en un 30 % anualmente.

El almacenamiento de la información y su tratamiento son importantes para cualquier sistema autoadaptable pero también lo son los mecanismos de interacción entre las personas y las máquinas y la interacción indirecta mediante la creación de aplicaciones sensibles al contexto. Con relación a los métodos de interacción Hombre-Máquina, hoy en día se dispone de numerosas interfaces de entrada-salida con las que un ser humano puede intercambiar información. Son dispositivos que van desde el clásico teclado o ratón, a las modernas pantallas táctiles, pasando por las cámaras web, los micrófonos, altavoces... y todos ellos suelen requerir estar delante del ordenador para llevarlas a cabo. La interacción con un sistema *cloud*, sin embargo, puede ser mucho más amplia, ya que al no disponer de un sistema físico localizado con el que llevar a cabo el intercambio de información, cualquier sensor puede actuar como interfaz de entrada, y cualquier pantalla o altavoz puede actuar como interfaz de salida. Así, por ejemplo, un lector de huellas en la puerta de una vivienda puede servir para identificar al usuario que accede a ella, y llevar a cabo una serie de acciones en consecuencia (poner música, encender la televisión mostrando los últimos movimientos de la bolsa...).

Estas acciones estarían orientadas a un hogar inteligente, pero con la cantidad de dispositivos inalámbricos presentes en la actualidad se puede extender más allá. Un móvil puede ponerse en silencio al detectar una red Wi-Fi asociada a nuestro lugar de trabajo, o ponerse en modo coche al detectar el Bluetooth del vehículo, o se puede usar la Wi-Fi para crear sistemas de localización [8][9], por ejemplo. La interacción en un sistema *cloud* no nos limita a un medio concreto, sino que en función de dónde necesitemos acceder a la información, podemos realizarlo de un modo u otro (por ejemplo, utilizando comandos de voz si estamos conduciendo, mediante una pantalla táctil que nos visualice pequeñas miniaturas

del concepto que buscamos si estamos en el trabajo o tecleando la búsqueda en un teclado sentados en el salón de casa). Incluso, en ciertas situaciones es necesario un claro entendimiento entre las partes, como por ejemplo entre médico y paciente, donde la claridad de la comunicación es vital, punto en el que la voz resulta ser el medio de transmisión más directo y sencillo para el humano [10].

En el *Handbook of Human-Computer Interaction* [6] se recoge una gran cantidad de estudios realizados acerca de la interacción entre humanos y máquinas, donde no sólo se contemplan dispositivos que se adapten al usuario en función de su edad, conocimientos, capacidad física u otros, sino que también hace mención a la interacción mediante sensores, interfaces tangibles y cognición aumentada, así como a la computación ubicua y portátil. Otros estudios han tomado como base esa información para realizar sus propios experimentos, obteniendo resultados sorprendentes como se dan en el libro *Research Methods in Human-Computer Interaction* [7].

Otro aspecto a tener en cuenta en un sistema es la ubicuidad y más concretamente la ubicuidad sensible al contexto. La computación ha ido evolucionando no sólo en la capacidad de procesamiento y velocidad de los equipos, sino también en la forma clásica de resolver los problemas. En sus inicios, los *mainframes*, grandes computadoras potentes y de gran volumen (llegando a ocupar habitaciones enteras) eran usados por multitud de personas, generalmente en una empresa o universidad, a través de terminales sencillos. Con el tiempo, el auge de los ordenadores personales descentralizó la computación, y hasta la llegada de internet los programas estaban aislados. Con la integración del *cloud computing* [10], el procesamiento de datos se mantiene en los ordenadores personales, a la vez que la información se encuentra distribuida y accesible por toda la red de computadoras integradas en el *cloud*. Gracias a ello, y al abaratamiento y miniaturización de los componentes electrónicos, se puede disponer de cualquier dispositivo conectado a la red, por pequeño o simple que pueda ser, lo que está llevando al fenómeno denominado **Internet de las Cosas** (*Internet of things*). Esto es posible gracias a la facilidad de obtener una conexión a internet, ya sea a través de cable o mediante sistemas inalámbricos como Wi-Fi o Bluetooth, el cual en la última versión incluso mantiene un bajísimo nivel de consumo de energía (ideal para pequeñas baterías).

En [11] ya proponían un sistema domótico para oficinas que se gestionaba de manera ubicua haciendo uso de un *cloud* para adaptar el sistema al contexto. Sin embargo esta computación ubicua presenta problemas cuando no se dispone de conectividad adecuada y en estos casos sería interesante posibilitar la ejecución local de parte de la funcionalidad llevada a cabo en la nube. En otras ocasiones puede ser de interés realizar la computación fuera de los dispositivos móviles para el ahorro de batería si se dispone de una conectividad adecuada. Es por eso que una computación sensible al contexto se hace necesaria en los sistemas actuales, de ahí que sea un parte relevante en el sistema planteado en este trabajo.

3. Arquitectura propuesta

La arquitectura propuesta toma como base un sistema multiagente que permite la integración de varios elementos de distinta naturaleza proporcionándoles un mecanismo de comunicación común. De esta manera, se pretende estructurar el sistema como una jerarquía de organizaciones virtuales, donde en cada nivel se pueden producir distintos eventos que sean desencadenados por agentes reactivos o por deliberativos, tras producirse un encadenamiento de reglas.

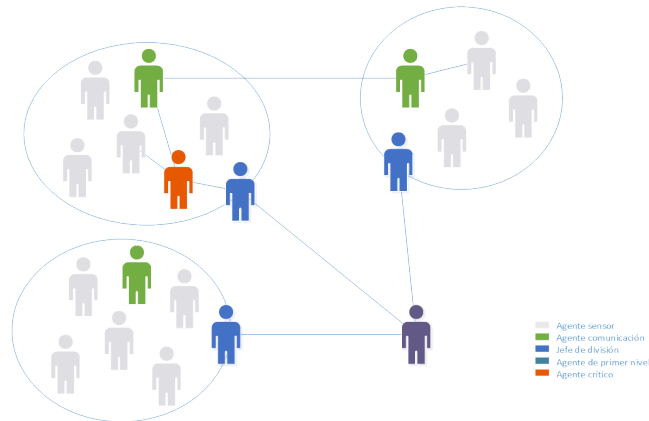


Figura 1. Estructura básica de las organizaciones virtuales

Respecto a la estructura jerárquica, se definen en primer lugar aquellos agentes reactivos cuya misión es alertar o reaccionar ante eventos críticos (agentes críticos, Figura 2), debiendo ser inmediatos en su actuación y no depender de ninguna orden superior. Para ello, se hará uso de agentes simples con trato directo en los sensores a monitorizar que pertenecerán a una división formada por ellos, el jefe de división asignado y los agentes que hayan de recibir el aviso de manera inmediata (actuadores en su gran mayoría) y que serán informados directamente por el agente crítico (excepto si pertenecen a otra división, caso en el que lo harán a través del agente comunicador). Además, estos agentes deberán notificar su existencia de manera periódica (enviando un *heartbeat* al jefe de división), con el objetivo de poder reaccionar a tiempo si por alguna razón perdieran la conexión con el sistema o el suministro energético.

Por otro lado, se encuentran los denominados como agentes móviles, cuya misión se centra en detectar la localización del usuario en cada momento e informar a sus superiores, con el fin de actuar en función de la situación física del usuario. Su funcionamiento se detallará más adelante. Estos agentes pertenecerán a una división específica, y sólo cambiarán su rol en caso de pérdida de conexión.

El resto de agentes se estructurará en una jerarquía de organizaciones, en la cual la base estará formada por aquellos agentes con tareas más sencillas y de

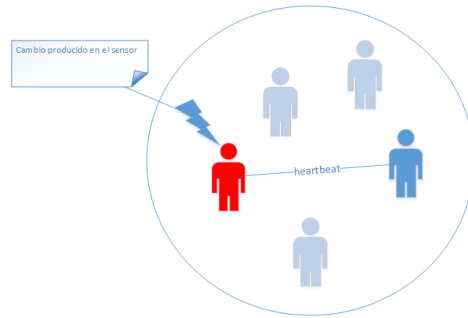


Figura 2. Organización virtual para los eventos críticos

más bajo procesamiento, como por ejemplo, aquellos encargados de monitorizar el estado de sensores que no entren en la categoría de críticos (reflejados en la Figura 2) y los responsables de la ejecución sobre los diversos actuadores. Estos agentes estarán agrupados en diversas organizaciones virtuales (divisiones) completamente independientes entre sí, y que tan sólo podrán comunicarse entre ellas a través de un agente “delegado” que transmita los mensajes entre su organización encargada y el resto (Figura 3).

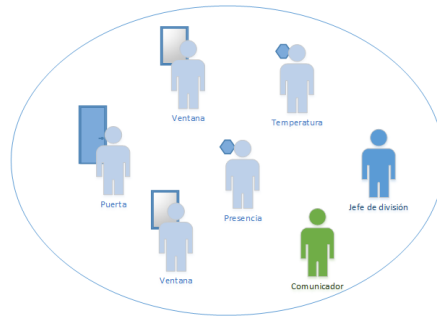


Figura 3. Organización virtual comprendida por algunos agentes de bajo nivel

Por encima de los agentes de bajo nivel se situarán los agentes denominados como “jefes de división”, cuya responsabilidad estará centrada en la supervisión de los agentes a su cargo y el consejo a los agentes superiores en función del estado de sus divisiones. Debido a la responsabilidad de estos agentes, deberá existir al menos uno en cada división susceptible de quedar incomunicada (pérdida de conexión, fallo en las comunicaciones, caída del sistema central, etc.), pudiendo operar de manera autónoma y local en esas situaciones, hasta que se pueda restablecer el control de todo el sistema. Para ello, dispondrán de una serie de reglas básicas con distinto orden de validez, donde la más alta puede quedar anulada por un agente de categoría superior.

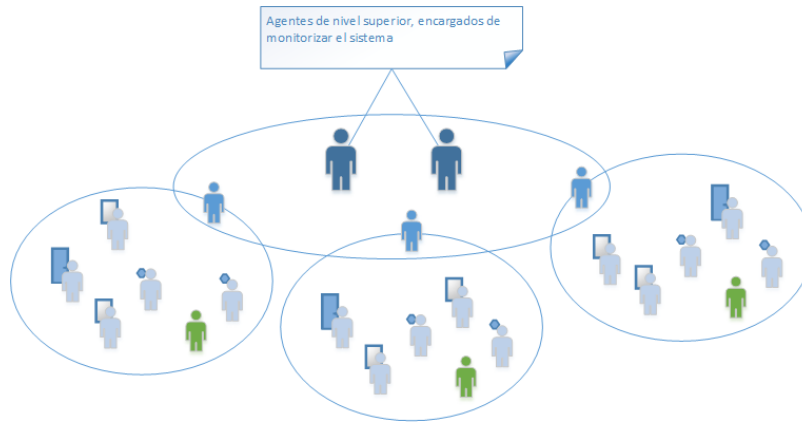


Figura 4. Organización virtual de nivel superior

En la Figura 4 se muestra cómo los jefes de división desempeñan dos roles, en función de la organización virtual en la que estén operando en ese momento. Por un lado, supervisan a los agentes de bajo nivel, y mantienen un registro de su estado. Por otro, interaccionan con los agentes de nivel superior, los cuales se encargan de mantener el sistema y hacen uso de los registros que recopilan los jefes de división. Este esquema permite delegar las responsabilidades si el sistema es muy amplio, generando divisiones intermedias compuestas por varios agentes de nivel X y un agente de nivel $X+1$.

3.1. Tipos de agentes en el sistema

Agentes reactivos Componen esta categoría los agentes de mayor urgencia en el sistema. Su función reside en la monitorización de elementos críticos, en los cuales cualquier fallo debe ser detectado y notificado para ser tratado con urgencia, independientemente del estado del sistema. Normalmente este tipo de agentes va asociado a la monitorización de sensores sensibles, como por ejemplo detectores de humo, de calor, escapes de gas, etc. Monitorizan directa y constantemente el sensor a su cargo (nunca más de uno) y controlan sus valores. Además, de manera periódica envían mensajes al sistema a modo de latidos, con el fin de confirmar su existencia y el funcionamiento de las comunicaciones. No se comunican con agentes de nivel superior, sino que sus avisos se publican en “salas” a los cuales se encuentran suscritos todos los agentes que deseen estar informados. Un ejemplo podría ser un agente que monitoriza un sensor de gas que, al detectar presencia de gas lanza la alerta en el tablón al cual está suscrito el agente encargado de cerrar la válvula de entrada del gas, de modo que puede realizar el cierre sin necesidad de que un agente de nivel superior tenga que ordenárselo.

El comportamiento de estos agentes es muy simple y sencillo, y sus reglas de actuación han de estar bien definidas para cada tipo de sensor. Dichas reglas se

basan en el esquema **si-entonces**, y se incorporan al agente en el momento de su creación.

En el caso de usar un motor de reglas, como por ejemplo Drools [21], la regla se podría expresar de la siguiente manera:

```
rule "Fire sensor"
when
    sensor:Sensor (value != 0);
    receptor:Tablon;
then
    receptor.publish(sensor.Id, sensor.value);
end
```

Este tipo de agentes no requiere de reglas más elaboradas, puesto que sus funciones han de ser simples y lo más rápidas posibles.

Agentes móviles Esta categoría especial de agentes se encuentra especializada en servicios de geolocalización. Pueden estar ubicados en vehículos, portátiles, lugares de trabajo. . . aunque su residencia más habitual hoy en día se encuentra en los *Smartphones*. La función principal reside en recolectar las ubicaciones que el usuario visita a lo largo del día, almacenarlas y llevar a cabo tareas acordes a dicha localización. También recoge información sobre el propio dispositivo en el que está alojado, ya sea la batería restante y las horas de uso en el teléfono móvil o el tiempo estimado de llegada en un vehículo, entre otros ejemplos. Junto con los datos de geolocalización almacena la información recogida con el fin de poder obtener decisiones basadas en históricos. Un ejemplo sencillo podría ser la entrada del agente en un área en la cual, generalmente, el usuario desconecta el sonido de su teléfono móvil. Puesto que este trabajo centra el uso de agentes móviles en *Smartphones*, los posibles casos focalizan la monitorización en el estado del audio del teléfono, su nivel de batería, sus rutas GPS y los eventos de la agenda.

En caso de perderse la comunicación con el sistema general (por fallo de cobertura, activación del ahorro de energía, etc.), este agente, tras unos intentos de reconexión y espera, pasará a operar en modo local, almacenando en una base de datos las localizaciones y los estados con el fin de volcarlos en el sistema una vez se restablezca la comunicación.

Jerarquía Los tipos de agentes anteriores, detallados en los casos anteriores operan de manera más aislada, pues la gran mayoría de ellos se agrupan en divisiones independientes intercomunicadas, donde cada una de ellas sigue su propio esquema de organización jerárquica. Este esquema facilita la creación automatizada de reglas de manera encadenada.

En el nivel inferior de categorías se encuentran los agentes sensores, cuya única función reside en registrar los cambios que detectan los sensores, almacenar la información y notificar el evento.

Los agentes sensores del mismo tipo se encuentran supervisados por un agente del tipo “jefe de división”, el cual mantiene un registro del estado de cada uno de ellos y manteniendo una visión global de esa categoría. Este agente aconsejará la activación de reglas utilizando el conocimiento obtenido de los agentes sensores a su cargo. Este grupo de agentes forma una división abstracta, pues no está sujeta a una distribución física de los componentes que monitoriza.

Por otra parte, se encuentra la división básica, la cual agrupa al conjunto de agentes sensores (y actuadores) de una ubicación concreta, junto con su jefe de división y el agente de comunicación.

Las distintas divisiones (u organizaciones virtuales) se supervisan y controlan por un conjunto de agentes especializados en distintos campos de acción (agentes de control de temperatura, gestión de eventos, automatización de tareas. . .) cuyo objetivo es la toma de decisiones en función del estado actual de las divisiones y su comportamiento pasado. A través del uso de un clasificador, entrenado con los datos recogidos del histórico, se puede mantener una hoja de ruta a la que, pasándole datos de los sensores y parámetros externos como la franja horaria o el día, recomiende las acciones a llevar a cabo.

Para dotar al sistema de mayor flexibilidad y ubicuidad, la información se almacena en la nube, con el objetivo de que pueda estar disponible para cualquier parte del sistema independientemente de la comunicación que pueda haber entre los distintos agentes.

Además, los agentes móviles deberán mantener una pequeña base de datos sobre la que volcar los datos en caso de pérdida de conexión, para posteriormente actualizarlos en el sistema cuando se vuelva a integrar en él.

4. Caso de estudio

El caso de estudio que se propone se centra en la gestión automatizada de una vivienda, con la finalidad de crear un ambiente que se adapte a las necesidades del inquilino en tiempo real. Para ello, el sistema controlará la temperatura de la vivienda en todo momento, actuando sobre ella en función de si hay gente en la vivienda o no, las condiciones climáticas externas, el tiempo restante para que haya alguien, etc. Además, adaptará los electrodomésticos a las necesidades del usuario en función de la franja horaria en la que se encuentre y la fecha. El sistema aprenderá las costumbres del usuario mediante la recolección de datos a través de los sensores y de los dispositivos móviles, con el fin de poder detectar patrones de comportamiento y predecir ciertas acciones.

4.1. Despliegue del sistema multiagente

Todos los sensores están gobernados por agentes simples, cuya misión será la de monitorizar su estado y almacenar cada cambio que se produzca en ellos. Cada uno de ellos pertenece, de manera independiente, a dos o más divisiones distintas, interpretando un rol en cada una de ellas. Por un lado, forman parte de una división que agrupa a todos los sensores de un mismo tipo (por ejemplo,

todos los agentes de presencia de la vivienda se agrupan en la división “presencia”) y a su vez cada uno forma parte de la división que engloba al conjunto de agentes de una pieza de la vivienda en concreto. Así, podemos conocer el estado general de un componente genérico de la vivienda o poseer una instantánea del estado de cada ubicación física de manera individual.

Como ejemplo, el funcionamiento de un agente sensor encargado de informar sobre el estado de una ventana, basa su funcionamiento en el siguiente bloque de código:

```
while(true)
{
    if(this.estado_anterior != this.estado)
    {
        this.estado = this.estado_anterior;
        notificarNuevoEstado(this.id, this.estado);
    }
}
```

Supervisando cada división se encuentra un agente que ocupa el rol de “jefe de división”, cuya labor es sondear el estado de sus agentes delegados, ordenar acciones en los actuadores, informar de manera periódica del estado de la organización y aconsejar la ejecución de reglas en función de dicho estado. Hay un jefe de división por cada habitación de la vivienda, además de jefes de división para cada categoría independiente: ventanas, puertas, electrodomésticos y presencia. En este último grupo se incluye un agente móvil que residirá en el *Smartphone* y cuya misión consistirá en informar al sistema dónde se encuentra el usuario cuando sale de la vivienda.

Bien cada 15 minutos o cuando el jefe de división recibe la notificación de cambio en algún sensor, introduce el nuevo estado en el gestor de reglas de usuario, el cual recibe el nombre de *Drools* y que es un sistema de gestión de reglas de negocio con un motor de inferencia de encadenamiento hacia adelante (también conocido como sistema de reglas de producción) y que está basado en el algoritmo *Rete* [22]. *Drools* utiliza un esquema de reglas basado en un antecedente, que marca la condición o condiciones necesarias para que se dispare una regla, y el consecuente, que contiene el conjunto de acciones a invocar en caso de cumplirse el consecuente. Está escrito en Java. La ventaja de utilizar este sistema para el gestor de reglas de negocio radica en que se utiliza en gran parte el lenguaje natural para definir los antecedentes y consecuentes de las reglas. De esta manera, a la hora de permitir que el usuario defina sus propias reglas personalizadas o anule la ejecución de otras automáticas, basta con pedirle en un primer paso qué eventos desea que se produzcan para la activación de la regla y, posteriormente en un segundo paso, qué acciones quiere llevar a cabo en las situaciones en las que dicha regla cumpla las condiciones adecuadas.

Un ejemplo de este tipo de reglas se puede mostrar para representar la siguiente situación: El usuario desea programar en el sistema una acción muy específica: cada vez que encienda la televisión, la iluminación del salón se debe

atenuar, con el fin de poder disfrutar de las películas. Pero además quiere que eso sólo ocurra cuando es por la noche, ya que durante el día le gusta disfrutar de la luz del sol. Para programar tal acción, el usuario accede al sistema a través de su dispositivo móvil u ordenador y solicita la opción “crear acción personalizada”. Al dar esa orden, el agente encargado de la gestión de acciones de usuario solicita a los jefes de división una lista de eventos o disparadores que puedan activarse en el sistema, una lista de acciones que puedan ejecutar y las muestra al usuario. Éste selecciona el evento “encender TV” (el cual es detectado por un agente que monitoriza el consumo de corriente de ese electrodoméstico). A continuación, se pide al usuario que seleccione un evento de entre los disponibles. Al seleccionar la acción de “luces”, especificar la intensidad, la pieza (o piezas si fuera necesario) de la habitación que se verá afectada y confirmar la acción, el agente de reglas produce una nueva regla de usuario que es enviada al jefe de división responsable del agente sensor escogido (o, si se hubieran seleccionado eventos de distintas divisiones, a los jefes responsables). Dicha regla quedaría definida en el lenguaje de *Drools* de la siguiente manera:

```
// El nombre lo adjudica el usuario,
// con el fin de que le sea fácil modificar
// o eliminar la regla cuando desee

rule ‘Modo cine en el salón’
when
    s1:SensorBinario(activo == true, tipo == ‘TV’);
    s2:SensorContinuo(valor > 60, tipo == ‘salon’);
    s3:SensorContinuo(valor < 30, tipo == ‘exterior’);
    a1:Actuador(tipo == ‘salon’);
then
    s2.setValor(valor); // Especificado por el usuario
end
```

Los agentes que utilizan árboles de decisión para la toma de acciones de manera autónoma (los de primer nivel y los jefes de sección) utilizan clasificadores basados en árboles de decisión. Un ejemplo de árbol de decisión generado para el control de seguridad es el que se muestra en la Figura 5:

4.2. Simulación de una jornada en día laboral

El caso de estudio que se presenta en esta memoria pretende demostrar el funcionamiento del sistema durante una jornada laboral. Los agentes de primer nivel se encuentran monitorizando la vivienda y consultando en el clasificador las acciones a realizar.

Para facilitar la explicación, en un primer momento se considera que el control de la temperatura devuelve un valor que no difiere en más de 2°C respecto a la temperatura actual de la vivienda, por lo que no se realiza acción ninguna. Del mismo modo, el control de seguridad determina que estando las puertas

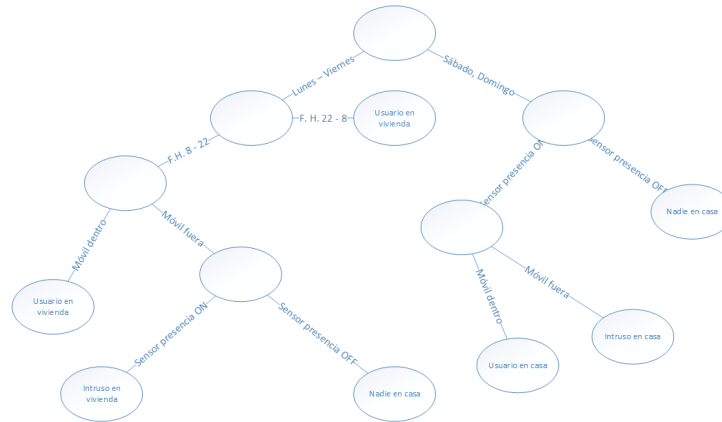


Figura 5. Árbol de decisión del controlador de seguridad

y ventanas cerradas, el agente móvil dentro de la vivienda y los sensores de presencia informando de que tan sólo hay un usuario, tampoco se lleva a cabo acción ninguna.

Respecto al gestor de acciones, posee una lista de eventos que ha sido previamente definida por el usuario. Para el caso de estudio, se ha programado el evento Despertador a las 7.00 y se ha programado que para ello encienda la luz de la habitación y active el hilo musical. A la hora programada, el agente gestor envía las órdenes al jefe de división de la habitación, el cual notifica a los actuadores de la lámpara y del hilo musical que se activen. Una vez hecho, el usuario ha de indicar que se finalice el evento mediante una acción directa, enviando la orden específica.

Cuando el usuario sale de la vivienda (considerando que lleva el *Smartphone* consigo) el sistema detectará que el agente móvil no se encuentra en la vivienda y los sensores de presencia darán negativo. El controlador de seguridad continúa consultando al clasificador cada 15 minutos, por lo que debería obtener la respuesta de “no hay nadie en casa”, por lo que entra en modo vigilancia y monitoriza los sensores de puertas, ventanas y presencia. En caso de detectar activación en alguno de ellos y seguir obteniendo la misma respuesta del clasificador, ejecutará la rutina de alarma.

Por su parte, el agente móvil, al detectar la pérdida de la conexión Wi-Fi, consultará a un clasificador propio, alojado en el *Smartphone*, utilizando como datos la pérdida de conectividad, última localización conocida, franja horaria, día de la semana, día del mes y mes actual. El clasificador debe devolver, de entre las posibles categorías, unas coordenadas GPS que el agente utilizará para sugerir la ruta de navegación (en este caso, al trabajo).

Durante la jornada, el usuario puede apuntar tareas y recados en su dispositivo móvil, con la hora a la que deben ser despachados al salir del trabajo, de manera que éste pueda calcular la ruta de vuelta, estimar el tiempo y avisar al sistema para que prepare la vivienda a la hora estimada de llegada (si tiene

que ir a comprar y hay tráfico, en lugar de volver sobre las 7 vuelve sobre las 9, el sistema introduce esa variación extra que se impone a la hora predicha de llegada por el clasificador de seguridad).

A la hora de salida, el sistema cargará la ruta de regreso, en función de si hay recados que atender, otros planes o simplemente la vuelta a casa.

Por su parte, el agente encargado del control de la temperatura, sondeará al clasificador para que informe sobre la temperatura óptima que debe haber en la vivienda. En caso de diferir en más de 2°C (valor fijado por el usuario) respecto a la temperatura actual, el agente solicitará la temperatura de manera individual a cada división física. Aquellas que difieran también en esa medida recibirán la orden de equilibrar la temperatura. Como excepción, si la temperatura es inferior a un umbral (definido por el usuario en los parámetros de configuración iniciales) el agente iniciará el sistema de calefacción de manera automática, en lugar de delegar en los jefes de división. De darse ese caso, sondeará el estado de las ventanas y, en caso de haber alguna abierta, notificará al usuario de la incidencia (pues eso contribuye a un gasto elevado de energía).

Para controlar la temperatura de manera local, cada jefe de división posee un clasificador propio al cual se le pasa la temperatura actual, la deseada, el estado de las persianas y ventanas y devuelve una de las siguientes categorías: abrir/-cerrar ventana, subir/bajar persiana, activar/desactivar aire acondicionado (en caso de disponer de ello).

5. Conclusiones y líneas futuras de trabajo

A lo largo de esta memoria se ha expuesto un modelo de organización virtual con un mecanismo de adaptación al contexto y que se sostiene sobre un entorno *cloud* que permite disponer de los datos de manera remota, sin necesidad de mantener copias redundantes de manera continua ni la necesidad de acceder a los registros para la realización de cualquier acción basada en históricos. Se han introducido los conceptos relacionados con sistemas multiagentes, su evolución en organizaciones virtuales, primando la adaptación y coordinación y diferentes técnicas que se pueden aplicar para llevar a cabo una distribución de tareas eficaz. Se ha explicado el modelo de datos enlazados y sus posibles aplicaciones, utilizando su filosofía para descentralizar el sistema y permitiendo el despliegue de agentes móviles que puedan mantener la interacción con el sistema en todo momento, además de comportarse de manera autónoma y adaptando su comportamiento a cambios en organizaciones virtuales que pudieran afectarles.

El uso de técnicas de minería de datos aplicados a conjuntos de *Big Data* ha permitido el tratamiento de la ingente cantidad de información que pueden proporcionar un conjunto no demasiado elevado de sensores a lo largo del tiempo, facilitando la creación de clasificadores y árboles de decisiones que permiten operar y predecir ciertos comportamientos en base a eventos anteriores que, por su naturaleza, por sí solos no proporcionan información alguna, pero que en conjunto permiten la extracción de características y rutinas que hacen que la plataforma esté en constante adaptación y evolución, acorde al comportamiento

del usuario. A medida que se usa la plataforma se puede lograr una mayor tasa de éxitos en las predicciones y en el comportamiento de los agentes.

Se ha realizado un estudio de las diferentes técnicas de tratamiento de datos, de los diversos sistemas de adaptación al contexto planteados en anteriores ocasiones y las diversas formas de interacción entre el ser humano y las máquinas, tanto de manera activa como pasiva. En este trabajo se ha hecho un mayor uso de la interacción pasiva del usuario con el sistema aprovechando la información que se genera día a día con nuestro comportamiento, además de la interacción a través de distintas interfaces existentes en la actualidad. Sobre todo se hace uso de la interfaz que proporciona un *Smartphone*, en parte por lo ampliamente extendidos que están en la sociedad actual y porque poco a poco se van convirtiendo en el centro de toda interacción activa hacia el mundo digital.

Una vez planteado el caso de uso, la arquitectura, el método de planificación y coordinación de la arquitectura y someter a evaluación un caso de estudio con datos simulados para comprobar la respuesta del sistema, se pretende continuar el trabajo con los siguientes puntos:

- Implantación de un conjunto de sensores en un entorno real reducido (una organización virtual física que abarque una habitación) y recopilar datos de los sensores durante un espacio de tiempo prolongado (de, al menos, un año de duración).
- Explorar otros métodos de extracción de características en técnicas de Big Data, con el fin de probar distintos algoritmos clasificadores y obtener los resultados que proporciona cada uno, someterlos a examen y evaluar cuál ofrece mejor rendimiento.
- Inclusión de más elementos sensores, con el fin de recopilar mayor información acerca de la rutina del usuario (como, por ejemplo, pulsómetros).
- Investigar nuevas formas de comunicación con la plataforma, de manera que no se centralice la interacción directa en el uso del Smartphone, sino que además incorpore métodos de “manos libres” (comunicación por voz).
- Finalmente se pretende investigar el funcionamiento de sistemas comerciales similares que han aparecido recientemente en el mercado, posteriores al planteamiento de esta propuesta (en especial sistemas de control del hogar) y las posibilidades de un planteamiento estándar de comunicación.

Bibliografía

- [1] P. Davidsson, “Agent based social simulation: A computer science view,” *Journal of artificial societies and social simulation*, vol. 5, no. 1, 2002.
- [2] L. Bruzzone and L. Carlin, “A multilevel context-based system for classification of very high spatial resolution images,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 44, no. 9, pp. 2587–2600, 2006.
- [3] E. Brown, R. Chang, H. Ellozy, J. Prager, and E. So, “System and method for hierarchically grouping and ranking a set of objects in a query context based on one or more relationships,” 1999. US Patent 5,875,446.
- [4] V. Kashyap and A. Sheth, “Semantic and schematic similarities between database objects: a context-based approach,” *The VLDB Journal—The International Journal on Very Large Data Bases*, vol. 5, no. 4, pp. 276–304, 1996.
- [5] M. M. Theimer, M. J. Spreitzer, M. D. Weiser, R. J. Goldstein, D. B. Terry, W. N. Schilit, and R. Want, “Selective delivery of electronic messages in a multiple computer system based on context and environment of a user,” 1996. US Patent 5,493,692.
- [6] A. Doan, N. F. Noy, and A. Y. Halevy, “Introduction to the special issue on semantic integration,” *ACM Sigmod Record*, vol. 33, no. 4, pp. 11–13, 2004.
- [7] N. R. Jennings and S. Bussmann, “Agent-based control systems,” *IEEE control systems*, vol. 23, no. 3, pp. 61–74, 2003.
- [8] M. Wooldridge and N. R. Jennings, “Intelligent agents: Theory and practice,” *The knowledge engineering review*, vol. 10, no. 02, pp. 115–152, 1995.
- [9] P. Zikopoulos, C. Eaton, *et al.*, *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media, 2011.
- [10] K. Crawford *et al.*, “Six provocations for big data,” 2011.
- [11] C. Lynch, “Big data: How do your data grow?,” *Nature*, vol. 455, no. 7209, pp. 28–29, 2008.
- [12] A. Berson and S. J. Smith, *Data warehousing, data mining, and OLAP*. McGraw-Hill, Inc., 1997.
- [13] “¿Qué es Big Data?”
- [14] “Cisco, internet será cuatro veces más grande en 2016, Artículo web.” (Última visita: 16/07/2014).
- [15] J. Rekimoto, T. Miyaki, and T. Ishizawa, “Lifetag: Wifi-based continuous location logging for life pattern analysis,” in *LoCA*, vol. 2007, pp. 35–49, 2007.
- [16] J. Biswas and M. Veloso, “Wifi localization and navigation for autonomous indoor mobile robots,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 4379–4384, IEEE, 2010.
- [17] O. L. Eyesan and S. R. Okuboyejo, “Design and implementation of a voice-based medical alert system for medication adherence,” *Procedia Technology*, vol. 9, pp. 1033–1040, 2013.

- [18] M. G. Helander, *Handbook of human-computer interaction*. Elsevier, 2014.
- [19] J. Lazar, J. H. Feng, and H. Hochheiser, *Research methods in human-computer interaction*. John Wiley & Sons, 2010.
- [20] S. S. R. Valenzuela and J. A. H. Terriza, “Servicios sensibles al contexto en sistemas de computación ubicua,”
- [21] M. Bali, *Drools JBoss Rules 5.0 Developer’s Guide*. Packt Publishing Ltd, 2009.
- [22] C. L. Forgy, “Rete: A fast algorithm for the many pattern/many object pattern match problem,” *Artificial intelligence*, vol. 19, no. 1, pp. 17–37, 1982.

Visualización de la evolución del software en sistemas de control de versiones

Rodrigo Olmo Rodríguez¹ y Roberto Therón Sánchez¹

¹Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España
{rodrigoolmo,theron}@usal.es

Resumen En esta memoria se detalla el proceso de investigación llevado a cabo en el ámbito de la visualización de datos y los sistemas de control de versiones de software. La cantidad de software desarrollado es mayor cada día, y los nuevos paradigmas y metodologías de desarrollo surgidos en los últimos años permiten a un mayor número de desarrolladores trabajar en paralelo sobre un mismo software. Conseguir visualizar la información contenida en un repositorio de software a lo largo del tiempo puede ayudar a entender su evolución, encontrar problemas y ayudar a tomar mejores decisiones.

1. Introducción

Los sistemas de control de versiones de software, conocidos como SVC (del inglés *System Version Control*) son herramientas que permiten almacenar y gestionar los cambios en un repositorio de software. Son ampliamente utilizados para archivos de texto (programas, documentación de software, gráficos vectoriales...) aunque también pueden ser utilizados con archivos binarios.

A medida que la cantidad de software desarrollado y utilizado se ha ido haciendo mayor con el paso de los años, las metodologías ágiles de desarrollo se han popularizado y muchos desarrolladores pueden trabajar en un mismo proyecto de software desde distintas partes del mundo gracias a Internet, se han desarrollado también distintos tipos de sistemas de control de versiones que han permitido responder a esos cambios.

El objetivo de este trabajo es aplicar técnicas de visualización de datos en sistemas de control de versiones de software. Permitir visualizar los datos contenidos en un repositorio de software puede ayudar a entender su evolución, encontrar problemas y tomar decisiones a diferentes niveles. Esto puede resultar útil tanto para un director de proyecto como para un desarrollador.

2. Sistemas de control de versiones de software distribuidos

Los sistemas de control de revisiones de software distribuidos (DRCS) también conocidos como sistemas de control de versiones descentralizados (DVCS)

facilitan aun más la tarea de trabajo simultáneo sobre un mismo proyecto entre varios desarrolladores distribuidos geográficamente.

Para conseguir esto se prescinde de la arquitectura cliente-servidor en la que existe un servidor centralizado que contiene el estado del proyecto considerado válido. En su lugar cada copia del proyecto es un servidor en potencia y puede actuar como tal.

Estos sistemas permiten realizar las tareas habituales (commit, log...) de manera mucho más rápida, ya que en lugar de realizarse directamente al servidor se realizan sobre la copia local.

El hecho de que cada copia del proyecto pueda actuar como servidor hace que la pérdida de datos sea mucho más improbable, ya que cada uno de los desarrolladores tendría una copia válida del proyecto que podría ser restaurada en cualquier momento.

La comunicación entre la copia local del repositorio de software y el servidor se realiza únicamente cuando se desea compartir y poner a disposición de los demás desarrolladores todos los cambios realizados.

3. Visualización de datos

La visualización de datos, la ciencia que trata de la representación visual de los datos, se ha definido como información que ha sido abstraída de alguna forma semántica, incluyendo atributos o variables para las unidades de información [1].

Adquiere su importancia por su capacidad de permitirnos “ver” cosas que no habían sido previamente entendidas en datos abstractos.

Colin Ware, experto en la psicología de la percepción, señala las potentes capacidades del sistema visual de los humanos para entender patrones y, por extensión, para entender datos. Como ejemplo de esto se puede ver en la Figura 1 la comparación entre dos métodos de representar datos sociográficos extraídos de cientos de regiones en Francia [2]. La imagen de la izquierda requiere buscar en la imagen el mayor y el menor valor, memorizarlos, y compararlos uno a uno. La representación de la derecha permite una comprensión cualitativa (existe algo importante en el noroeste del país).

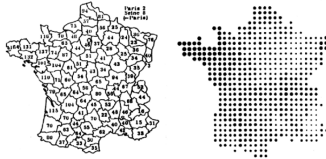


Figura 1. Comparación entre dos métodos para representar un conjunto de datos sociográficos de Francia

4. Aspectos técnicos

La herramienta desarrollada para este trabajo, *Mercurize*, utiliza varios lenguajes de programación para la obtención, proceso y visualización de los datos.

Debido a que el sistema de control de versiones utilizado se trata de Mercurial[3] y éste software está escrito en el lenguaje de programación Python [4], se ha optado por utilizar ese mismo lenguaje para el desarrollo de la aplicación web que extrae los datos del repositorio de software, los procesa, y genera la página web donde está contenida la visualización. Teniendo en cuenta que Mercurial dispone de una API en Python para acceder a todos los datos y funcionalidades de los repositorios de software, esto permite que la extracción de datos pueda ser realizada con mayor rapidez y de manera nativa.

Para el desarrollo de la aplicación web se ha utilizado *Flask*, un framework de aplicaciones web escrito en *Python*, y que pretende llevar la flexibilidad y sencillez de dicho lenguaje al desarrollo de aplicaciones web.

Para la visualización de datos se ha optado por utilizar Javascript y la librería D3.js. D3.js (*Data Driven Documents*) es una librería que permite manipular documentos basados en datos utilizando tecnologías web estándar (como HTML, SVG y CSS) y permitiendo la inspección/manipulación de una representación nativa: el estándar *the standard document object model* (DOM) [5].

El hecho de utilizar estos componentes estandarizados permite aprovechar el potencial de los navegadores web modernos. También permite simplificar el desarrollo, ya que todos los componentes podrán ser inspeccionados y depurados desde las herramientas de desarrollo incluidas en la mayoría de navegadores web.

D3.js incluye funciones Javascript que permiten:

- Seleccionar elementos.
- Crear objetos SVG.
- Dar estilo a estos objetos.
- Añadir transiciones y efectos.

El formato de datos utilizado para serializar los datos una vez que han sido extraídos y procesados ha sido JSON (JavaScript Object Notation) [6]. Se trata de un formato de datos estándar derivado del lenguaje de programación Javascript (aunque se trata de un formato de datos independiente del lenguaje) y creado como alternativa XML.

Para la maquetación de los elementos y el estilo general se ha utilizado la librería de CSS Twitter Bootstrap [7].

Como editor de texto tanto para el código escrito en Python como el escrito en Javascript se ha utilizado SublimeText 2 [8].

5. Visualización de sistemas de control de versiones de software

Se han utilizado diferentes tipos de visualizaciones y de elementos visuales para este problema.

- **Streamgraph.**
Utilizada por primera vez en el año 2008 en el periódico norteamericano *The New York Times* [9], permite representar datos que tienen una variable temporal, y se visualiza el cambio producido en otras variables a lo largo del tiempo.
- **Zoom.**
Debido a que el número de datos que se ha de visualizar es muy elevado se puede perder detalle de las variaciones a nivel concreto. Por otra parte, si la visualización únicamente permitiera ver un conjunto reducido de datos, se estaría perdiendo la información global.
Utilizar zoom permite poder visualizar el contexto global, y acercar y desplazar la visualización hacia aquellas zonas que sean de interés. Gracias a la transición entre un estado y otro no se pierde el contexto.
- **Escala.**
Se permite dentro de la aplicación desarrollada poder alternar entre una escala lineal y una escala de raíz cuadrada.
No se ha utilizado la escala logarítmica debido a su imposibilidad para representar el 0.
- **Interacción.**
Existen diferentes tipos de interacción dentro de la aplicación desarrollada que permiten ampliar la información obtenida y mejorar la percepción visual.
 - **Búsqueda en el texto del commit.**
Se permite buscar por una o varias palabras clave que estén contenidas en el texto del commit. Esto resaltará aquellos commits que cumplan esa condición.
 - **Información del commit.**
Al pasar el ratón por encima de un commit se visualiza en la parte inferior la información más relevante de ese commit.
 - **Cambio de escala.**
Permite cambiar la escala entre las dos indicadas anteriormente.
 - **Cambio de variable en el eje X.**
Se permite cambiar entre Revisión y Fecha, en función de si se desean visualizar los datos como una sucesión periódica que depende del identificador de commit, o como una serie temporal.
- **Transiciones.**
Todos aquellos cambios que sean resultado de una interacción por parte del usuario (cambio de zoom, cambio de escala, cambio de variable en el eje X...) se visualizarán utilizando transiciones, y de esta forma permitiéndose que el usuario no pierda el contexto de lo que estaba visualizando.

6. Conclusiones

La investigación llevada a cabo en este trabajo, centrada principalmente en la visualización de la evolución del software en los sistemas de control de versiones,

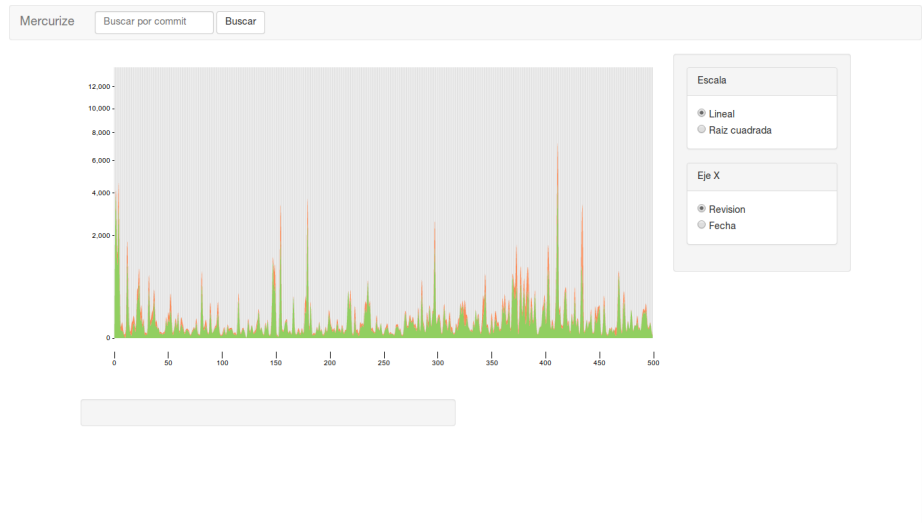


Figura 2. Visión general de Mercurize

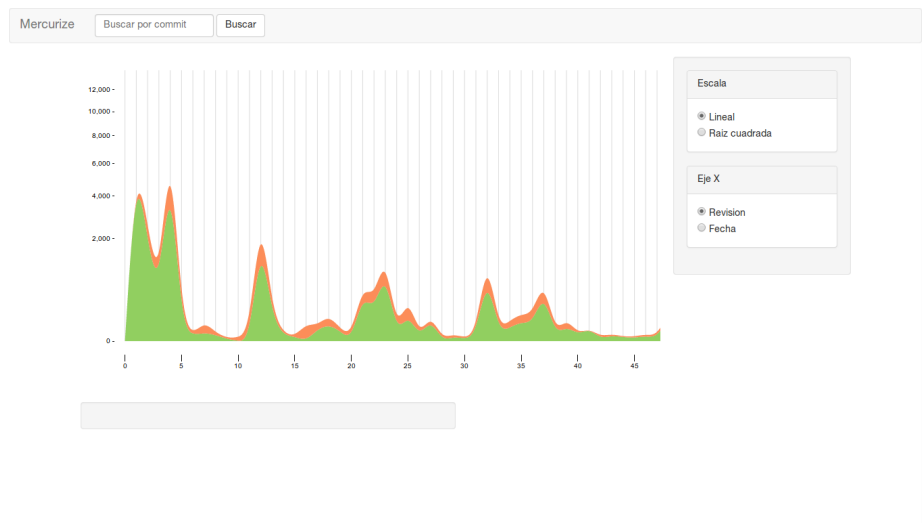


Figura 3. Uso del zoom

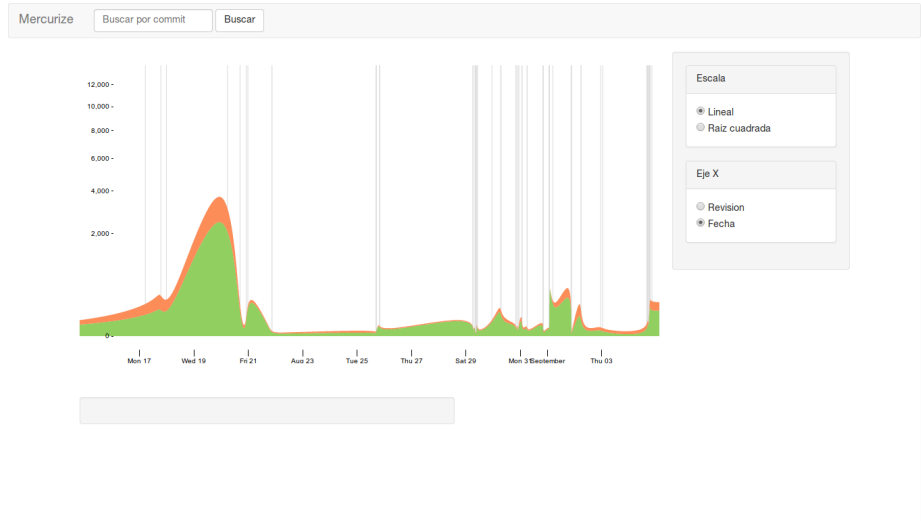


Figura 4. Uso del eje X escalado por fecha

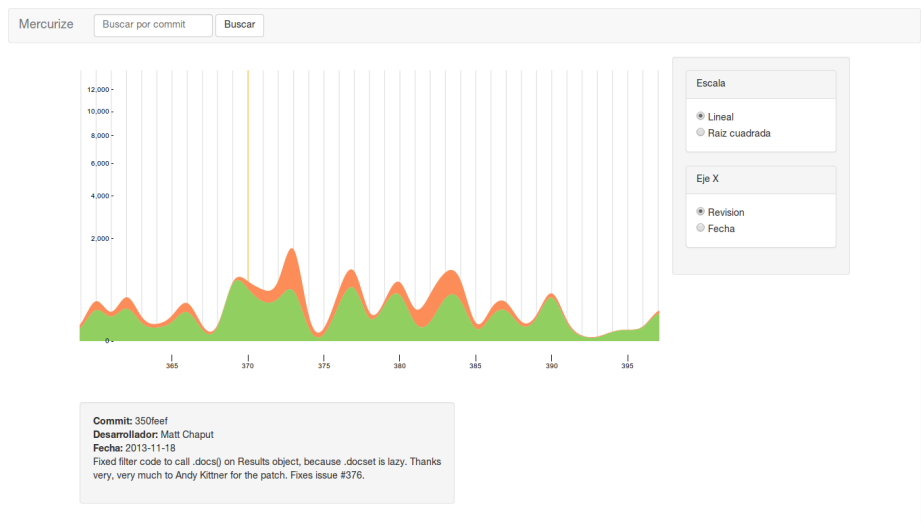


Figura 5. Información sobre un commit al interactuar con la visualización

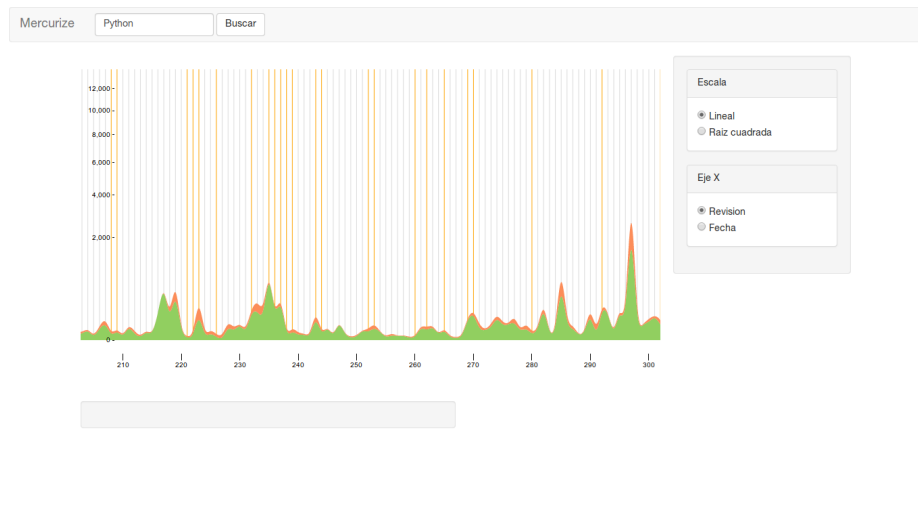


Figura 6. Búsqueda de la palabra “Python” y visualización de los resultados.

ha permitido proponer un tipo de visualización (streamgraph) que resulta útil para la comprensión de los repositorios de software.

El hecho de que la herramienta desarrollada (*Mercurize*) sea accesible desde un entorno web facilita su uso a los posibles usuarios, ya que no existiría necesidad de instalar ningún software adicional excepto en el servidor donde se ejecutara la aplicación.

Bibliografía

- [1] M. Friendly and D. J. Denis, “Milestones in the history of thematic cartography, statistical graphics, and data visualization,” *Seeing Science: Today American Association for the Advancement of Science*, 2008.
- [2] J. Bertin, *Semiology of Graphics - Diagrams, Networks, Maps*. ESRI, 2010.
- [3] M. Mackall and S. Consulting, “Json.”
- [4] Python Software Foundation, “Python.”
- [5] M. Bostock, V. Ogievetsky, and J. Heer, “D³ data-driven documents,” *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, 2011.
- [6] ECMA International, “Mercurial SCM.”
- [7] Bootstrap, “Bootstrap.”
- [8] Skinner, Jon, “Sublime Text.”
- [9] L. Byron and M. Wattenberg, “Stacked graphs - geometry & aesthetics,” *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1245–1252, 2008.

Sistema Multiagente eficiente para la extracción de características y análisis de imágenes

Isabel Martín Rodríguez¹, Juan Francisco De Paz Santana¹ y Belén Pérez Lancho¹

¹Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España
{ysamr, fcofds, lancho}@usal.es

Resumen El objetivo de este trabajo es crear un sistema que permita analizar la significación de las diferencias entre un conjunto de clasificadores. El análisis estadístico consiste en el cálculo de la estimación de la bondad del clasificador, que permite ofrecer la capacidad de predicción de dicho clasificador, seguido de un estudio de significatividad, que permite discriminar cuál de las dos hipótesis consideradas, la hipótesis nula y la hipótesis alternativa, es la verdadera mediante un conjunto de tests paramétricos o no paramétricos. Para ello se plantea el desarrollo de una arquitectura multiagente a la que se le ha aplicado un caso de estudio concreto en el que se analizan las estimaciones obtenidas por varios clasificadores de un conjunto de imágenes a las que se les ha realizado diferentes tipos de corte.

Keywords: Sistema Multiagente, Análisis estadístico, tests paramétricos, tests no paramétricos

1. Introducción

Los algoritmos de extracción de patrones presentan un elevado coste computacional, por lo que es necesario realizar un estudio de dichos algoritmos determinando cuál es el más óptimo. Un concepto fundamental en el ámbito de los métodos de clasificación son los diversos criterios para la evaluación de los clasificadores. Entre ellos se encuentra la estimación de la bondad de un clasificador que permite efectuar una medición sobre la capacidad de predicción del modelo generado a partir de un clasificador. Para estimar la bondad de un clasificador se pueden utilizar diferentes pruebas como son el índice Kappa [1], que mide la contingencia de la predicción con la clase real, o la tasa de acierto, que mide el número de términos correctamente reconocidos por un algoritmo respecto al total. Tradicionalmente, en la ciencia de la computación, los diferentes procesos, procedimientos o algoritmos se realizan mediante representaciones gráficas o mediante valores numéricos pero no se suele realizar un estudio sobre si las diferencias son significativas o no. El estudio de la significatividad de las diferencias es un aspecto importante en la rama de la ciencia de la computación y más particularmente en áreas como la minería de datos. Así estos estudios serían de relevancia en apartados como el manejo de clasificadores para el estudio de

la capacidad de clasificación, en sistemas de predicción para el análisis del error o para el manejo de cualquier técnica, por ejemplo para la comparación de los tiempos de ejecución.

Herramientas como Weka [2] permiten comparar diferentes clasificadores a partir de la realización de técnicas como *10-fold cross validation* pero sólo proporcionan la media de varios índices por las 10 iteraciones. Este valor no sirve para determinar si las diferencias son o no significativas puesto que además de la media hay que tener en cuenta aspectos como la varianza. Existen varios programas como SPSS [3] capaces de realizar test estadísticos que proporcionan un resultado preciso pero que sin embargo no ofrecen el proceso completo de evaluación de los clasificadores, el cual abarca desde un preprocesamiento de datos hasta un análisis de los resultados.

El procedimiento que se utiliza para poder comparar si dos o más algoritmos se comportan de forma similar es el test estadístico, que permite discriminar cual de las dos hipótesis consideradas, la hipótesis nula y la hipótesis alternativa, es la verdadera al aplicar dicho test a un conjunto de experimentos. Los test estadísticos que se pueden aplicar se dividen en paramétricos y no paramétricos, dependiendo de si el conjunto de datos cumple o no una serie de características [4] [5]. Algunos de los tests que se pueden usar son: el análisis de la varianza ANOVA, con su correspondiente prueba post hoc en el caso de igualdad de medias. La prueba t de Student [6] para dos muestras independientes en la que se evalúa la diferencia significativa entre las medias de dos grupos, los cuales corresponden con los datos obtenidos de las estimaciones de dos algoritmos de extracción de patrones. Las pruebas de Kruskal-Wallis [7], alternativa no paramétrica de ANOVA, y Mann-Whitney [8][9], la alternativa no paramétrica de la prueba t de Student.

En este trabajo se propone el desarrollo de una arquitectura multiagente que permita realizar validaciones estadísticas a partir de un conjunto de estimaciones de varios clasificadores, incorporando diferentes test en función de las características de las poblaciones muestrales. Como caso de estudio, se propone realizar una implementación a partir de una arquitectura propuesta basada en AUML [10] [11] que permita incorporar los servicios necesarios para obtener un conjunto de datos, resultado de la aplicación de dos o mas algoritmos de clasificación de sexo y de edad respectivamente, y realizar un análisis estadístico. Cabe destacar el aspecto dinámico de la plataforma que permite modificar el comportamiento del agente principal, el cual se encarga de gestionar a los demás agentes y de gestionar los datos, a partir de las sentencias declaradas en un archivo XML. Dichas sentencias definen el comportamiento del agente describiendo un flujo de tareas en las que debe acceder a diferentes servicios obteniendo un conjunto de datos que se encargara de proporcionar a los diferentes agentes que conforman el sistema.

En este trabajo se utiliza la plataforma multiagente JADE [12], basada en los estándares FIPA [13] para definir los diferentes protocolos de comunicación y un vocabulario genérico para los agentes, para realizar el desarrollo de un sistema multiagente, y librerías Java para realizar los test estadísticos como *Apache*

Commons Math.

El artículo está estructurado en cuatro partes. En la primera parte se realiza un estudio de los sistemas multiagente, se describen las medidas para la comparación de los clasificadores y los tests estadísticos utilizados. En la segunda parte se describe la arquitectura propuesta capaz de obtener un conjunto de datos de al menos dos algoritmos distintos a un conjunto de imágenes y de realizar los cálculos necesarios para obtener una comparación entre clasificadores. La tercera parte presenta un caso de estudio, que demuestra el funcionamiento de la arquitectura propuesta utilizando JADE como plataforma multiagente, junto con los resultados obtenidos. Por último se muestran las conclusiones obtenidas y el trabajo futuro.

2. Estado del arte

2.1. Agentes y sistemas multiagente

Con la Inteligencia Artificial Distribuida (IAD) surgen los orígenes de la tecnología de agentes. Un agente es un sistema informático situado en algún entorno, dentro del cual actúa de forma autónoma y flexible para así cumplir sus objetivos. Un agente es capaz de percibir dicho ambiente a través de sensores, evaluar tales percepciones y tomar un conjunto de decisiones que le permitan actuar sobre el medio a través de efectores [14].

En general un sistema multiagente [15][16] está formado por múltiples agentes autónomos que muestran las siguientes características:

- Los agentes son incapaces de resolver un problema por sí mismos, es decir, necesitan de la ayuda de otros para conseguir sus metas.
- No hay un sistema de control global que gestione las interacciones.
- La información está descentralizada.
- La computación es asíncrona.

2.2. Técnicas de comparación

Para estimar la precisión de los clasificadores, una de las técnicas más utilizadas es la de *cross validation*. Esta técnica consiste en dividir los datos aleatoriamente en un conjunto de entrenamiento, que a su vez se divide en dos conjuntos disjuntos llamados conjunto de estimación y conjunto de validación, y un conjunto de test [17] [18] [19] [20].

2.3. Medidas para la comparación de clasificadores

Las medidas para la comparación de clasificadores permiten obtener la precisión que el clasificador comete en el conjunto de test.

La tasa de acierto relaciona el número de aciertos obtenidos por el clasificador con el número total de casos mediante el cociente: número de aciertos/número total de casos, proporcionando un porcentaje de aciertos.

El índice Kappa [1] relaciona el acuerdo que exhiben los observadores, más allá del debido al azar, con el acuerdo potencial también más allá del azar. La estimación por el índice de kappa sigue la ecuación:

$$k = \frac{P_0 + P_e}{1 - P_e} \quad (1)$$

2.4. Métodos de análisis de datos

No se puede determinar a partir del cálculo de una medida de comparación mediante validación cruzada si hay diferencias significativas entre clasificadores, y por tanto no se puede saber cuál es más óptimo. Es por ello que hay que realizar una validación sobre un conjunto de medidas de comparación utilizando métodos de análisis estadístico.

Según [4] en la comunidad científica a día de hoy es cada vez más demandado el uso de tests estadísticos apropiados en función de las propiedades de los datos obtenidos. Estos tests estadísticos nos permiten determinar si los resultados obtenidos son significativamente diferentes en los algoritmos de comparación y si las conclusiones son compatibles con el experimento llevado a cabo.

De todas las distribuciones posibles que puedan seguir unos datos, la distribución normal es la más interesante desde el punto de vista estadístico, pues reúne unas propiedades que han hecho posible que a partir de ella se desarrollaran numerosos métodos de análisis de datos. En ella, los valores cercanos a la media son los más abundantes y a medida que se van alejando de la media, los datos presentan una frecuencia cada vez menor. La distribución normal posee una serie de características:

- Corresponde a variables cuantitativas continuas.
- Se caracteriza por dos medidas: media y desviación típica.
- Es unimodal.
- Es simétrica alrededor de la media. Por tanto, media, mediana y moda coinciden.
- Tiene forma acampanada, sin un pico excesivo.
- El área bajo la curva es 1.

Se han desarrollado numerosos test estadísticos que permiten realizar pruebas de contraste de hipótesis a partir de una distribución normal, los cuales reciben el nombre de pruebas paramétricas. Sin embargo, no siempre los datos obtenidos se ajustan a la distribución normal, por lo que para hacer pruebas de contraste de hipótesis se recurre a la estadística no paramétrica. En las pruebas de contraste de hipótesis, las diferentes pruebas estadísticas utilizan la llamada hipótesis nula (H_0). La hipótesis nula siempre presupone que no existen diferencias entre los grupos o asociación entre las variables. El nivel de significación (α) es el parámetro que cuantifica el error que se está cometiendo al aceptar los resultados. Concretamente, lo que indica es la probabilidad de que se rechace la H_0 siendo cierta. Para tomar una decisión respecto a cuál sea la hipótesis ‘verdadera’, el investigador fija el nivel máximo de error que se permite. En general, se ha fijado

por convenio el umbral de $\alpha = 0,05$ como valido.

Los métodos paramétricos se basan en ciertas suposiciones acerca de los parámetros de la población μ , que representa la media de la población, y σ^2 , que representa la varianza de la población [4] [5].

- Homogeneidad de varianzas (homocedasticidad).
- Independencia de los errores.
- Normalidad [21] [22].

La prueba t-Student [6] permite calcular la diferencia significativa entre las medias de dos grupos con distribución normal. El estadístico t para varianzas iguales se obtiene a partir de la siguiente ecuación:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{S_{x_1x_2} * \sqrt{1/n_1 + 1/n_2}} \quad (2)$$

siendo $S_{x_1x_2}$ la desviación estandar combinada la cual se calcula mediante la siguiente ecuación:

$$S_{x_1x_2} = \sqrt{\frac{(n_1 - 1)S_{x_1}^2 + (n_2 - 1)S_{x_2}^2}{n_1 + n_2 - 2}} \quad (3)$$

donde $S_{x_1}^2$ y $S_{x_2}^2$ son las cuasivarianzas muestrales correspondientes, n_1 y n_2 los tamaños de las muestras, $n - 1$ es el número de grados de libertad para cada grupo y $n_1 + n_2 - 2$ es el número de grados de libertad utilizados para la prueba de significancia.

El analisis de la varianza (ANOVA) permite calcular la hipótesis de un conjunto de medias de varios grupos. Se trata de una generalización del test t para dos muestras. La hipótesis nula de la prueba ANOVA de un factor es:

$$H_0 : \mu_1 = \dots = \mu_r = \mu$$

$$H_1 : \exists i, j / \mu_i \neq \mu_j$$

Las pruebas ANOVA requieren de la distribución F [6][20]. Suponiendo que las poblaciones tienen la misma varianza σ^2 , el estadístico de prueba F es la razón o el cociente de los siguientes dos estimados de σ^2 :

1. La variación entre muestras (con base en la variación entre medias muestrales).
2. Variación dentro de muestras (con base en las varianzas muestrales).

Cuando tras realizar la prueba de analisis de la varianza se obtiene un rechazo de la hipótesis nula, existe diferencia significativa entre al menos dos medias de los grupos. Pero no se puede concluir a partir de una prueba ANOVA qué media en particular es diferente de las demas. Existen varias pruebas, llamadas pruebas de comparaciones múltiples, que utilizan pares de medias pero hacen ajustes para superar el problema de tener un nivel de significación que aumenta conforme se incrementa el número de de pruebas individuales. Una de ellas es la prueba de Bonferroni [6]. El test de Bonferroni se basa en calcular una prueba t para cada par de muestras:

$$t = \frac{\bar{x}_i - \bar{x}_j}{\sqrt{CM(error) * (1/n_1 + 1/n_2)}} \quad (4)$$

donde:

$$CM(error) = \frac{\sum (n_i - 1) * S_i^2}{N - k} \quad (5)$$

Siendo N el número de valores de todas las muestras y k el número de muestras. Después de calcular el valor del estadístico de prueba t para un par específico de muestras, se debe calcular el valor t crítico o el valor P . Para calcular el valor P se debe utilizar el estadístico de prueba t con $gl = N - k$, pero multiplicandolo por el número de pares de muestras diferentes posibles.

La prueba de Mann-Whitney [8] es el equivalente no paramétrico de la prueba t de Student para la diferencia de dos medias cuando las muestras son independientes pero no cumplen los requisitos de homocedasticidad o normalidad. Para realizar el contraste se ordenan de forma creciente todas las observaciones en una sola serie asignando rangos de 1 a $n_1 + n_2$. Posteriormente se obtiene R_i que es la suma de los rangos de cada una de las muestras. El estadístico de prueba U de Mann-Whitney [9] es el siguiente:

$$U_i = n_1 n_2 + \frac{n_i(n_i + 1)}{2} - R_i \quad (6)$$

donde $i = 1, 2$.

Bajo el supuesto de que la hipótesis nula sea cierta, la media y la desviación típica de U son:

$$\mu_U = \frac{n_1 n_2}{2} \quad (7)$$

$$\sigma_U^2 = \frac{n_1 n_2 (n_1 + n_2 + 1)}{12} \quad (8)$$

El estadístico de prueba es el valor Z :

$$Z = \frac{U - \mu_U}{\sigma_U} \quad (9)$$

El contraste de Kruskal-Wallis, propuesta por William Henry Kruskal (1919-) y W. Allen Wallis (1912- 1998) en el artículo "Use of ranks in one-criterion variance analysis" publicado en el "Journal of American Statistics Association" en 1952 [7], es la alternativa no paramétrica del método ANOVA. El único requisito que deben cumplir los grupos es la aleatoriedad en la extracción de las muestras. La hipótesis nula de la prueba de Kruskal-Wallis es:

H_0 : Las k medianas son todas iguales

H_1 : Al menos una de las medianas es diferente

Para cada observación se le asigna el rango según el orden que ocupa la observación en el conjunto total de los datos. El estadístico de contraste de Kruskal-Wallis H se calcula como:

$$H = \frac{12}{N(N + 1)} \sum_{i=1}^C \frac{R_i^2}{n_i} - 3(N + 1) \quad (10)$$

siendo C el número de muestras, n_i el tamaño de una muestra, N el tamaño total de todas las muestras y R_i la suma de los rangos de la muestra i -ésima.

Si cada muestra tiene al menos cinco observaciones, la distribución de H es aproximadamente una distribución Chi-Cuadrada con $k - 1$ grados de libertad.

3. Descripción de la arquitectura propuesta

El objetivo principal es crear un sistema multiagente que permita realizar un conjunto de test estadísticos para obtener el clasificador mas óptimo. Para realizar dicho sistema se propone una arquitectura base en la cual los diferentes métodos estadísticos estén repartidos en varios agentes junto con un agente que actúa de manager de todo el sistema (figura 1).

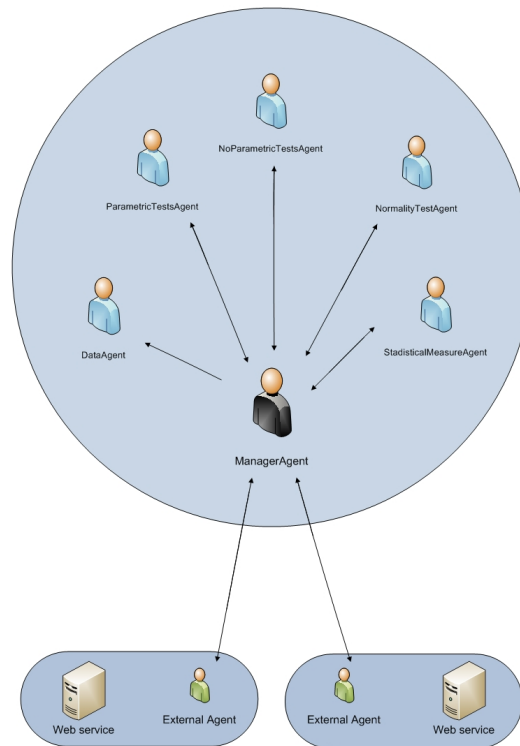


Figura 1. Arquitectura del sistema multiagente

Como puede apreciarse en la figura 1, el sistema esta compuesto por un conjunto de agentes, con roles bien definidos, que comparten información y servicios. El analisis estadístico de datos implica un proceso en el que cada agente ejecuta su tarea con la información disponible en cada momento. El agente principal se encarga de interactuar con los agentes externos a través de los servicios que

estos ofrecen permitiéndole obtener un conjunto de datos resultado de varios clasificadores de imágenes. Los pasos que el sistema debe seguir en el proceso pueden apreciarse en el diagrama de flujo de la figura 2.

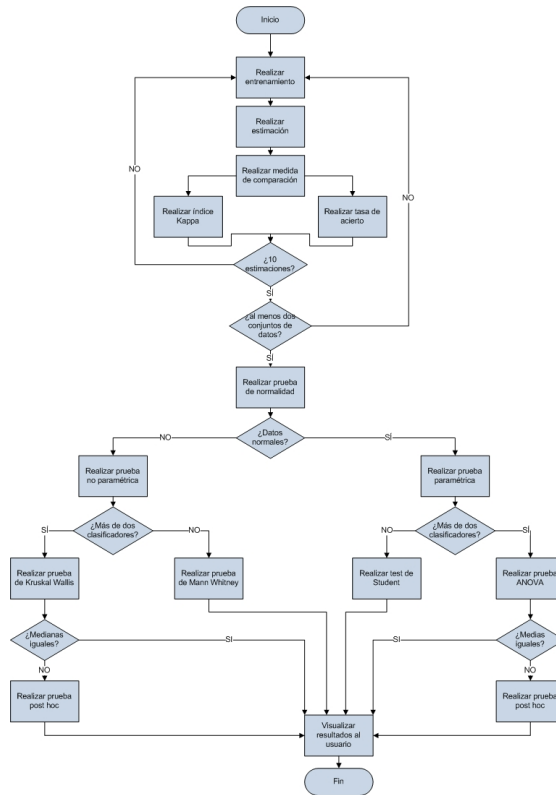


Figura 2. Diagrama de flujo del sistema multiagente

Estos pasos no son estáticos pues se pretende que el ManagerAgent tenga un comportamiento dinámico que pueda ser modificado. Este comportamiento se pretende plasmar en un archivo XML cuyo contenido describa los pasos a realizar y más concretamente el flujo de datos que debe realizarse entre el ManagerAgent y los demás agentes. La estructura que se pretende obtener en dicho archivo XML consta de un conjunto de acciones las cuales indicarán al ManagerAgent a qué servicios debe acceder para obtener un flujo de datos que tenga una estructura coherente y qué datos debe mostrar al usuario final.

Una vez obtenidos los datos de la estimación de un clasificador, se procede a realizar un factor de bondad el cual permite efectuar una medición sobre la capacidad de predicción de dicho clasificador. Este proceso se repite mediante una técnica de validación cruzada hasta obtener un conjunto de factores de bondad

de un clasificador. Igualmente se debe repetir el proceso hasta tener dos o mas conjuntos de factores de bondad, uno por cada clasificador. Sobre estos conjuntos se realizara una prueba paramétrica o no paramétrica, en función de si estos datos siguen una distribución normal, para determinar si existen diferencias significativas entre ambos clasificadores. Finalmente los resultados son mostrados al usuario. Se pretende que el proceso realizado por el agente principal sea dinamico de manera que puedan ser añadidos nuevos agentes que proporcionen diferentes test estadísticos, o modificados o eliminados los ya existentes. Durante el analisis se descompone el proceso descrito anteriormente de manera que las tareas queden distribuidas en roles. Los agentes que se han obtenido se describen a continuación:

- **ManagerAgent**: es el agente principal encargado de obtener los datos sobre las estimaciones de los clasificadores. Realiza el proceso descrito en el diagrama de flujo obtenido en la figura anterior.
- **StadisticalMeasureAgent**: se encarga de realizar el factor de bondad a partir de las estimaciones obtenidas. Para ello calcula el índice Kappa o la tasa de acierto sobre el conjunto de datos.
- **NormalityTestAgent**: es el encargado de realizar la prueba de normalidad sobre los datos, los cuales estan compuestos por un conjunto de índices de bondad. La prueba de normalidad debe comprobar si los datos siguen una distribución normal, si existe homogeneidad de varianzas y si los datos son independientes.
- **ParametricTestAgent**: este agente se encarga de realizar las pruebas paramétricas de ANOVA y t de Student en el caso de que los datos cumplan los requisitos de normalidad.
- **NoParametricAgent**: si los datos no cumplen los requisitos de normalidad, se deberán realizar las pruebas de Kruskal Wallis o Mann Whitney, las cuales son realizadas por este agente.
- **DataAgent**: se encarga de mostrar los resultados finales al usuario.

Dentro de la etapa de analisis se pretende comprender el sistema y su estructura sin entrar en detalles de implementación, lo que se reduce a definir el conjunto de roles del sistema aplicando la metodología Gaia [23]:

- **Buscador**: obtiene una lista de servicios proporcionados por los módulos externos, igualmente obtiene una lista de los clasificadores que estos módulos utilizan.
- **ManejoAgentes**: gestiona los agentes del sistema controlando su actividad y planificando un orden.
- **Visualizador**: obtiene los resultados finales y los procesa para ser visualizados al usuario.
- **CalculadorMedidasComparación**: calcula una medida de comparación, índice Kappa o tasa de acierto, sobre un conjunto de datos.
- **CalculadorNormalidad**: realiza la prueba de normalidad sobre los datos.
- **CalculadorParamétrico**: realiza los tests paramétricos.
- **CalculadorNoParamétrico**: realiza los tests no paramétricos.

Para realizar los modelos en la fase de diseño se utiliza la metodología AUML (Agent Unified Modeling Language) que permite el desarrollo de sistemas multi-agente partiendo de los conceptos base de UML [10] pero sin estar completamente restringida a estos. Los agentes obtenidos son los mismos que los de la etapa de analisis, pero con un nivel mas de abstracción, en el que se definen los servicios y las capacidades que estos van a ofrecer:

- El ManagerAgent tendra la capacidad de realizar los procesos definidos en el diagrama de flujo de la figura 2, los cuales se plasman en un archivo XML. No va a ofrecer ningún servicio, pues su función principal es obtener datos de los módulos externos y enviarlos a través de mensajes a los diferentes agentes.
- El StadisticalMeasureAgent es capaz de realizar dos medidas, tasa de acierto e índice Kappa, para calcular el factor de bondad de un conjunto de datos, por lo que ofrecera dos servicios, uno para cada medida. El ManagerAgent descrito anteriormente, podra acceder a cualquiera de los servicios enviando un mensaje que contenga el conjunto de datos sobre los que se quiere realizar el calculo.
- El NormalityTestAgent ofrecera un único servicio correspondiente al calculo de la normalidad, la cual implica tres tareas que deben describirse como capacidades. A través de ese servicio podra obtener los datos que le envie el ManagerAgent y enviar una respuesta sobre la normalidad de los mismos.
- El ParametricTestAgent es el encargado de realizar las dos pruebas paramétricas del sistema. Ademas de tener la capacidad de realizar las pruebas paramétricas, realiza una prueba post hoc en el caso de igualdad de medias en la prueba ANOVA. Ofrece dos servicios, uno por cada test paramétrico.
- El NoParametricTestAgent es similar al ParametricTestAgent, ya que ofrece dos servicios pero en este caso para los dos test no paramétricos disponibles, el test de Kruskal Wallis y el test de Mann Whitney.
- El DataAgent ofrece un único servicio a través del cual podra recibir los resultados finales pudiéndolos mostrar al usuario.

4. Resultados

A partir de la implementación de la arquitectura propuesta, se procede a evaluar un conjunto de datos obtenidos del módulo que realiza estimaciones sobre el sexo de un conjunto de imagenes. Dichas estimaciones se realizan utilizando dos clasificadores distintos que son los algoritmos Fisherfaces y una red neuronal. Dentro de cada clasificador, las imagenes procesadas se recortan de cuatro maneras diferentes al realizar el entrenamiento y la estimación. Para evaluar los diferentes clasificadores, se debe utilizar una técnica de *cross validation*, en este caso se utiliza un *5x2 cross validation* el cual consiste en dividir la muestra en dos grupos (A y B según las tablas 1 y 2) y realizar el proceso de entrenamiento-estimación 5 veces. En las tablas 1 y 2 se muestra la tasa de acierto de cada clasificador con los cuatro cortes realizados.

Cuadro 1. Tasa de acierto para el clasificador fisherfaces aplicando 5x2 cross validation

Corte	A1	B1	A2	B2	A3	B3	A4	B4	A5	B5	Media
k0,5w2	0.903	0.882	0.901	0.886	0.893	0.875	0.886	0.884	0.907	0.882	0.889
k0,5w2,2	0.897	0.895	0.899	0.880	0.897	0.884	0.884	0.847	0.893	0.884	0.886
k0,35w2,2	0.884	0.888	0.901	0.875	0.886	0.871	0.880	0.858	0.886	0.858	0.880
k0,35w2,4	0.893	0.897	0.884	0.890	0.884	0.886	0.873	0.899	0.899	0.893	0.890

Cuadro 2. Tasa de acierto para el clasificador RNA aplicando 5x2 cross validation

Corte	A1	B1	A2	B2	A3	B3	A4	B4	A5	B5	Media
k0,5w2	0.867	0.895	0.837	0.871	0.865	0.860	0.854	0.860	0.897	0.817	0.862
k0,5w2,2	0.856	0.888	0.869	0.882	0.895	0.862	0.845	0.886	0.847	0.856	0.869
k0,35w2,2	0.877	0.895	0.862	0.858	0.884	0.852	0.856	0.860	0.843	0.832	0.862
k0,35w2,4	0.865	0.877	0.884	0.832	0.841	0.837	0.869	0.852	0.841	0.888	0.858

Una vez se han obtenido los resultados, se debe realizar un estudio que obtenga si existe diferencia significativa entre los clasificadores. Se van a realizar dos estudios:

1. El primer estudio consistirá en comparar los cuatro cortes dentro de cada clasificador, para obtener el mejor corte dentro de cada clasificador en caso de que lo haya.
2. El segundo estudio consistirá en realizar comparaciones por pares de los mismos cortes para cada clasificador, para obtener si un corte es más óptimo que el otro entre los dos clasificadores.

En el primer estudio, primero se cogen todos los valores de los cuatro cortes correspondientes al primer clasificador, es decir, los cortes correspondientes a fisherfaces. Tras realizar una prueba de normalidad se obtiene que aunque los valores son independientes y hay homogeneidad de varianzas, con un valor $P = 0,3629$ según el test de Barlett, los datos no siguen una distribución normal, pues se obtiene un valor P de 0,012, se rechaza la hipótesis nula debido a que $P < 0,05$. Por tanto se debe realizar una prueba no paramétrica para más de dos muestras, la prueba de Kruskal Wallis. Una vez realizada la prueba de Kruskal Wallis los valores obtenidos son:

$$H = 4,1234$$

$$P = 0,2484$$

Como $P > 0,05$ entonces se acepta la hipótesis nula de igualdad de medianas, lo que quiere decir que no existe diferencia significativa entre el conjunto de muestras a la hora de usar cualquiera de los cuatro cortes dentro del mismo clasificador. Al igual que se ha hecho un análisis con todos los cortes dentro del clasificador Fisherfaces, se debe realizar lo mismo con los cortes del clasificador RNA. Tras realizar la prueba de normalidad se obtiene que las muestras cumplen los requisitos paramétricos, ya que son independientes, se cumple la homogeneidad de varianzas con un valor $P = 0,8422$, y siguen la distribución normal, con un valor $P = 0,44$. Por tanto se deberá realizar la prueba paramétrica para tres

o mas muestras de ANOVA. Los resultados de la prueba de ANOVA son los siguientes:

$$F = 0,4252$$

$$P = 0,7361$$

Como $P > 0,05$ se acepta la hipótesis nula de igualdad de medias, lo que quiere decir que no existe una diferencia significativa a la hora de usar cualquiera de los cuatro cortes dentro del mismo clasificador.

Para el segundo estudio se realizan comparaciones de los mismos cortes de los dos clasificadores, con el fin de obtener si hay uno de los dos que sea mas óptimo en un clasificador que en otro. La tabla 3 muestra los valores obtenidos a lo largo del analisis estadístico.

Cuadro 3. Comparación por pares de los mismos cortes para cada clasificador aplicando 5x2 cross validation

Corte	Normalidad	Homocedasticidad	t-test	Mann Whitney
k0,5w2	0.003323	0.02	-	0.0065017
k0,5w2,2	0.164039	0.622	0.0325466	-
k0,35w2,2	0.16	0.342	0.034933	-
k0,35w2,4	0.0033	0.0101	-	0.0009

Si los valores P obtenidos en los tests de Student y de Mann Whitney respectivamente son inferiores que el nivel de significación ($\alpha = 0,05$), se puede concluir que los cortes presentan diferencias significativas al usarlos en un clasificador o en otro. Según la tabla 3, existen diferencias significativas en todos los pares de cortes.

5. Conclusiones

El trabajo de investigación realizado pretende presentar nuevas perspectivas en el ambito del analisis estadístico utilizando sistemas multiagente. En este sentido, el sistema multiagente creado permite validar estadísticamente un conjunto de datos procedentes de varios clasificadores para obtener si existe alguno que sea mas óptimo sobre los demas. Esta arquitectura podría utilizarse no sólo para clasificadores de imagenes como los utilizados en el caso de estudio, sino para cualquier conjunto de clasificadores que se deseen analizar.

Los resultados obtenidos muestran que, mediante el analisis estadístico, se pueden hacer afirmaciones sobre si existen o no diferencias significativas para un conjunto de clasificadores. Sin embargo, todos los test estadísticos descritos en este trabajo deben ser considerados como unas pruebas heurísticas aproximadas, en vez de pruebas rigurosamente correctas.

5.1. Líneas futuras de investigación

A continuación se muestran posibles líneas de investigación futuras para el presente trabajo:

- Los resultados obtenidos en el aspecto dinámico del AgenteManager no han sido tan favorables como se esperaba, por lo que como trabajo futuro se pretende optimizar el flujo de trabajo realizado mediante dicho agente a través de un archivo XML.
- Añadir nuevos tests estadísticos como pueden ser el test de Friedman [4], prueba de Wilcoxon [24] y el test de McNemar [19]. Además de diferentes pruebas post hoc como el test de Holm [24] o el test de Hochberg [24].
- Mejorar la precisión de los datos obtenidos por los test estadísticos de manera que los resultados sean de una fiabilidad mayor.
- Realizar mejoras en el aspecto gráfico de la interfaz implementada por el DataAgent.

Bibliografía

- [1] J. Cohen, “Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit,” *Psychological Bulletin*, vol. 70, no. 4, 1968.
- [2] “Weka 3: Data mining software in java.”
- [3] “Ibm, software spss.”
- [4] J. Luengo, S. García, and F. Herrera, “A study on the use of statistical tests for experimentation with neural networks: Analysis of parametric test conditions and non-parametric tests,” *Expert Systems with Applications*, vol. 36, no. 4, pp. 7798–7808, 2009.
- [5] D. J. Sheskin, *Handbook of parametric and non-parametric statistical procedures*. CRC Press, 2003.
- [6] M. F. Triola, *Estadística*. Pearson Education, décima edición ed., 2009.
- [7] W. Kruskal and W. Wallis, “Use of ranks in one-criterion variance analysis,” *Journal of American Statistics Association*, 1952.
- [8] J. Manju and C. Priebe, “A data-adaptive methodology for finding an optimal weighted generalized mann-whitney-wilcoxon statistic,” *Computational Statistics & Data Analysis*, vol. 51, pp. 4337–4353, 2007.
- [9] B. Mann and D. Whitney, “On a test of whether one of two random variables is stochastically larger than the other,” *Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, 1947.
- [10] O. M. G. Inc, “Unified modeling language specification. version 1.4,” 2001.
- [11] G. Booch, J. Rumbaugh, and I. Jacobson, “El lenguaje unificado de modelado,” 1999.
- [12] “Java agent development framework.”
- [13] “Foundation for intelligent physical agents.”
- [14] S. Labidi and W. Lejouad, “De l’intelligence artificielle distribuée aux systèmes multi-agents,” 1993.
- [15] N. Sridharan, “Report on the 1986 workshop on distributed artificial intelligence,” *The AI Magazine*, pp. 75–85, 1986.
- [16] M. Wooldridge and N. Jennings, “Intelligent agents: Theory and practice,” *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.
- [17] F. Herrera, C. Hervás, J. Otero, and L. Sánchez, “Un estudio empírico preliminar sobre los tests estadísticos más habituales en el aprendizaje automático,” *DigitalTres*, pp. 403–412, 2004.
- [18] M. Stone, “Cross-validation: a review,” *Series statistics*, vol. 9, no. 1, pp. 127–139, 2007.
- [19] T. G. Dietterich, “Approximate statistical tests for comparing supervised classification learning algorithms,” *Neural Computation*, pp. 1895–1923, 1997.
- [20] E. Alpaydin, “Combined 5x2cv f test for comparing supervised classification learning algorithms,” *Neural Computation*, vol. 11, pp. 1885–1892, 1998.
- [21] S. S. Shapiro and M. B. Wilk, “An analysis of variance test for normality (complete samples),” *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.

- [22] H. W. Lilliefors, “On the kolmogorov-smirnov test for normality with mean and variance unknown,” *Journal of the American Statistical Association*, vol. 62, no. 318, pp. 399–402, 1967.
- [23] M. Wooldridge, N. Jennings, and D. Kinny, “The gaia methodology for agent-oriented analysis and design,” *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 3, pp. 285–312, 2000.
- [24] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” *Swarm and Evolutionary Computation*, vol. 1, pp. 3–18, 2011.

Minería de datos aplicada al análisis de datos RNA-seq

Óscar González Velasco¹ y Luis Antonio Miguel Quintales²

¹ Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España

`oscargv@usal.es`

² Instituto de Biología Funcional y Genómica (IBFG), Universidad de Salamanca/CSIC. Calle Zacarías González, 2. 37007, Salamanca, España

`lamq@usal.es`

Abstract. Since its initial application several years ago, parallel cDNA sequencing, more commonly known as RNA-seq, has helped to develop the way of making transcriptome profiling. Studies using this method have already altered our view of the extent and complexity of eukaryotic transcriptomes organisms. We can summarize the concept of RNA-seq as a high throughput sequencing technology that uses the capabilities of next-generation sequencing, capable of reading a snapshot of RNA presence and quantity from a genome at a given moment in time, obtaining the RNA transcripts in a cell (transcriptome). But these advantages have also come with arising new problems: RNA-Seq requires increasingly complex computational methods. These computational challenges can be listed into three main categories: mapping reads to a genome, transcriptome reconstruction and expression quantification. In addition, the enormous amount of generated data, like values of the gene expression, differential analysis and read counting has shown an additional problem: infer valid knowledge from this vast set of raw data. Computing techniques such as data mining, pattern recognition and statistical analysis can be used to obtain valuable information from these datasets. This article shows the different steps to carry out a full RNA-Seq analysis, detailing the bioinformatic pipeline necessary to achieve satisfactory results: raw data is obtained using RNA-Seq sequencing, this data is later dumped to a file. Our challenge will be to process this dataset obtained from the RNA-Seq analysis. To realize it, firstly we will align RNA-Seq reads using a reference genome, then we will count the obtained reads. Later we will calculate the gene expression levels (a value that tells us how each gene is being expressed), finally we will carry out a differential expression analysis over the data from the previous stages in order to compare the samples with each other. Also we will apply some data mining techniques and explain the different steps that will lead us to complete a successful data mining study, using the biological data obtained in the previous stages.

Keywords: Bioinformatics, Data Mining, RNA-Seq, Massive sequencing

1. Introduction

RNA-seq (RNA Sequencing) is a high throughput sequencing technology that uses the capabilities of next-generation sequencing, capable of reading a snapshot of RNA presence and quantity from a genome at a given moment in time, obtaining the RNA transcripts in a cell (transcriptome) [1]. These data is later dumped to a raw file, which will be processed using bioinformatics techniques.

2. RNA-Seq: Next Generation Sequencing

Over the past 20 years successive advances in biology and in particular in the field of genomics due to the inclusion of high advanced technologies and computing science have thrown new light over life science, whose consequence has been a deep understanding of the way life works. RNA-Seq uses recently developed deep sequencing technologies [2]. In general, a population of RNA (total or fractionated) is converted to a library of cDNA fragments (complementary DNA (cDNA) is the DNA synthesized from a messenger RNA (mRNA)), these fragments include adaptors attached to one or both ends. Thus, each RNA molecule is then sequenced in a high-throughput manner to obtain short sequences from one end (single-end sequencing) or both ends (pair-end sequencing).

2.1. Analysis Steps of a RNA-Seq Experiment

Results and quality of the assembled transcriptome will be directly affected by the chosen design of the RNA-Seq experiment [3], including the construction of the cDNA fragments library, the selection of the next-generation sequencer used, and the protocol used in the data analysis phase. In general, RNAseq experiments follow a series of steps leading to a similar protocol, though variants depending on the experiment are usual. At the end of the process a file containing the RNA-Seq reads will be obtained, this data will be later processed using several bioinformatic techniques to retreat valuable information.

1. Firstly, RNA is isolated from a previous selected sample which, depending on the type of RNA to be proled, can be puried to enrich for mRNAs, microRNAs etc prior to preparing an RNA library.
2. The preparation of the library involves steps such as reverse transcription to cDNA, PCR amplification and many more.
3. When all these previous steps are successfully carried on, a raw datale containing the shortreads of the RNA that were able to be read in the sample will be obtained.

2.2. Data Analysis (pre and post processing)

Once the data has been generated, a file containing the sequenced reads will be available for further processing; this step is what we will cover in this paper.

In the data analysis phase these short reads are pre-processed to remove possible sequencing errors, besides other artefacts [4]. Since the implantation of high-throughput sequencing technologies, RNA-Seq has faced several informatics challenges. Some of the most important are: the development of efficient methods to analyze, store and process large amounts of sequenced data and identify and remove low quality reads. Once RNA-Seq data has been generated, is needed to pre and post process the sequencing output data to obtain information and knowledge. A basic RNA-Seq data processing pipeline will have the following steps:

1. Align or assemble reads to reveal transcription structure.
2. Count reads, estimate abundance and gene expression.
3. Post-processing: differential analysis, gene significance, data mining, gene ontology and more.

Aligning Reads Once high-quality reads have been obtained, the first task of data analysis is to map the short reads from RNA-Seq to a reference genome, however, it pose particular challenges, because reads are short (36 to 125 base-pares [5] [1]), error rates are considerable and many reads span exon-exon junctions. To address assembly we will be using an aligner which makes use of the algorithmic approach known as unspliced read aligner. This method aligns reads to a reference genome without allowing any large gaps. It exist two types [6] [7] of unspliced read aligners: seed methods and Burrows-Wheeler transform methods.

One of the aligners that we will be using in our experiment: Bowtie [8] falls in the category of Burrows-Wheeler transform methods, which compact the genome into a data structure that is very efficient when searching for perfect matches [9]. If these classes of aligners are configured to accept mismatches, the performance of Burrows-Wheeler transform methods decreases exponentially with the number of mismatches. A special case is the TopHat2 [10] aligner that we will be using for our experiment, which make use of a two-step process. First, it map reads continuously to the genome making use of the unspliced read aligner Bowtie. Second, unmapped reads are split into shorter segments and aligned independently.

Estimating Abundance and Transcriptome Reconstruction Expression counting is not trivial for transcriptomes with extensive alternative splicing: transcripts often share some exons, causing uncertainty as to which transcript each read belongs to [11]. The accuracy and precision of gene expression counting are influenced by cDNA library construction methods, sequencing technologies and data pre-treatment techniques. Transcriptome reconstruction is a difficult computational task for three main reasons First, gene expression spans several orders of magnitude, with some genes represented by only a few reads. Second, reads originate from the mature mRNA (exons only) as well as from the incompletely spliced precursor RNA (containing intronic sequences), making it difficult

to identify the mature transcripts. Third, reads are short, and genes can have many isoforms, making it challenging to determine which isoform produced each read.

Post-processing: Differential Analysis, Data Mining and more When transcriptome reconstruction and counting have been done we need some post-processing in order to infer valuable information from the huge quantity of output data. Some analysis, like differential expression, will help us to compare samples and conditions. Others like datamining will help us to find patterns of expression among sets of genes which, with the help of other tools, can tell us the shared functionality or proteins linked with a set of distinctive genes.

3. RNA-Seq Pipeline

To successfully align and count reads, estimate abundance, perform transcriptome reconstruction, and thus obtain final results, we will need a wide variety of tools and software, which is commonly named as the RNA-Seq pipeline [12].

3.1. Pipeline Tools

A group of tools were selected to join this pipeline from all of the existing bioinformatic tools, including: Bowtie [8] an ultrafast, memory-efficient short read aligner geared toward quickly aligning large sets of short DNA sequences (reads) to large genomes, published by the Johns Hopkins University. It works best when aligning short reads to large genomes, though it supports arbitrarily small reference sequences. TopHat [10], a program that aligns RNA-Seq reads to a genome in order to identify exon-exon splice junctions, published by the Johns Hopkins University. It is built on the ultrafast short read mapping program Bowtie explained above. Though we can achieve aligning with the bowtie tool, Tophat can also find splice junctions without a reference annotation. Cufflinks [13] (and its secondary tools: cuffdiff, cuffmerge and more) is a tool for assembling transcripts, estimating their abundances, and testing for differential expression and regulation in RNA-Seq samples. It accepts aligned RNA-Seq reads and assembles the alignments into a parsimonious set of transcripts. Cufflinks then estimates the relative abundances of these transcripts based on how many reads support each one, taking into account biases in library preparation protocols. SAMtools [14] is a set of utilities for interacting with and post-processing short DNA sequence read alignments in the SAM/BAM format. Integrated Genome Browser [15] is an easy-to-use, highly customizable genome browser that can be used to view and explore genomic data and annotations, especially RNA-Seq and ChIP-Seq data sets.

The main characteristic that we were seeking was the speed, since alignment and transcriptome reconstruction is a high demanding process, and the equipment used for this study was very limited, the more appropriate aligner seemed Bowtie, which compact the genome into a data structure that is very efficient

when searching for perfect matches (9). In contrast, when only the reference transcriptome of a distant species is available (not our case), ‘seed methods’ (such as SHRiMP and Stampy) can result in a large increase in sensitivity. The choice of Tophat was not only because the main engine of this aligner is build on the ultra-fast aligner Bowtie, but also because it creates a database of all possible combinations of splicing junctions within a locus and then align to this database the reads that failed to align to the genome, resulting in a more precise alignment. The typical RNA-Seq pipeline that I would be using here will be as follows:

- Download the genome (fasta file) and the gene transcripts file, create the genome index using bowtie.
- Download RNA-Seq data (SRA File) from the chosen dataset, convert the raw data file into a fastq file using sraTools.
- Map sequenced reads (fastq file) to the target genome using TopHat.
- Transform the aligned reads (bam file) into a bedgraph file in order to view the genomic expression graphically in the IGB.
- Calculate read counts, gene expressions and statistics using Cufflinks.
- Realize a differential expression analysis between the genes of the targeted samples and its replicates.
- Apply data mining and other statistical analysis and visualize the results.

Example of an experiment involving two conditions: reads are first mapped to the genome using TopHat. Reads for each biological replicate are mapped independently. These mapped reads are provided as input to Cufflinks, which produces one file of assembled transfrags for each replicate. The assembly files are merged with the reference transcriptome annotation into a unified annotation for further analysis. This merged annotation is quantified in each condition by Cuffdiff, which produces expression data in a set of tabular files. These files are examined, the genes identified by Cuffdiff as differentially expressed are selected, and finally we apply some datamining techniques and we visualize all the results using visualization techniques, to facilitate the understanding. (Please Note: original image was taken from nature protocols [12]).

4. RNA-Seq EXPERIMENT DATASET

An experiment was chosen in order to carry a real RNA-Seq data analysis, the aim is to apply the RNA-Seq pipeline on it and to post-process the data results, trying to reproduce the original experiment results. The complete article whose name is The Yeast Snt2 Protein Coordinates the Transcriptional Response to Hydrogen Peroxide-Mediated Oxidative Stress [16] is available at <http://mcb.asm.org/content/early/2013/07/16/MCB.00025-13>. The dataset of this article can be public accessed using GEO (Gene Expression Omnibus, <http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE42983>), the available raw files correspond to not-processed RNA-Seq data.

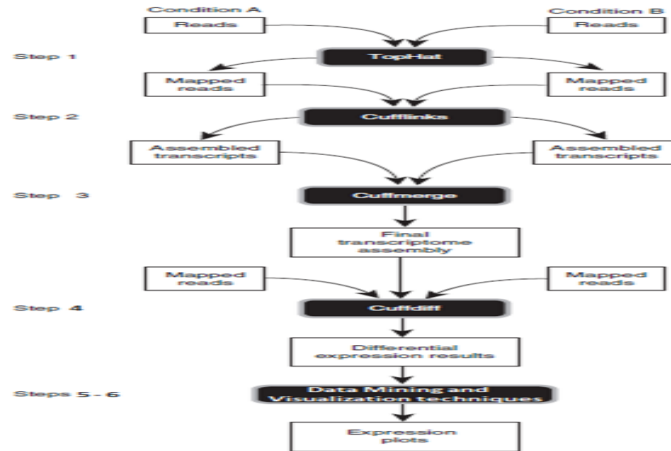


Fig. 1. RNA-Seq Illustration Diagram [12]

4.1. Experiment Goals

The main key the experiment tries to determine is whether Snt2 is involved in regulating the transcriptional response to oxidative stress (oxidative stress refers to an imbalance between the production of free radicals and the ability of the biological body to counteract or detoxify their harmful effects through neutralization), thus multiple lines of evidence point to a role for Snt2 in the oxidative stress response or possibly in the ESR (environmental stress response genes, a group of genes involved in numerous types of stress). The experiment was done using *Saccharomyces cerevisiae*, a species of yeast. Also, Snt2 is believed to be physically associated with Ecm5 and Rpd3. To better understand how Snt2 and Ecm5 might function during stress, RNA-sequencing (RNA-seq) analysis was used to identify target genes that required Snt2 for proper expression changes in H_2O_2 stress, including genes involved in the ESR. In the same way, some other genes are studied, like the case of both cell wall mannoprotein genes FIT1 and DAN1. Snt2 and Ecm5 are believed to be related with cell wall regulation functions, the original experiment tried to determine the possible relation between these two cell wall genes and Snt2/Ecm5.

5. Analyzing the Dataset

In this section I describe the process to obtain different output data (Gene Expression Level, Read Count, Differential Expression), visualize these data, as well as applying some data mining and statistical analysis over the output data, obtained from the RNA-Seq raw data (dataset corresponding to the experiment explained in section: 4 RNA-Seq EXPERIMENT DATASET: [16]).

5.1. Obtaining *Saccharomyces Cerevisiae* Genome and GTF File

Both, GTF file and Genome were obtained from: <http://www.ensembl.org/info/data/ftp/index.html> NOTE that is recommended that both files come from the same source, as it is possible that positions of transcripts between the genome and the genes differ slightly, causing an unmatched result. The Gene transfer format (GTF) is a file format used to hold information about gene structure (position of the gene in the genome and its name).

5.2. Creating the Index

Once we have downloaded the *Saccharomyces cerevisiae* genome, we need to build the genome index. To build the index from the fasta file we will use the Bowtie tool: bowtie-build. The goal is to create a valid index in order to use as a reference genome and use it as a target to achieve alignment from reads. To create the index of the reference genome, follow these steps:

1. Download the reference genome
2. Move the reference genome .fasta file into the Bowtie install directory
3. Build the index using bowtie. Using the command line type:

```
$ ./bowtie-build
Schizosaccharomyces_pombe.ASM294v2.21.dna.toplevel.fa
Schizosaccharomyces_pombe
```

Where:

- a) Schizosaccharomyces_pombe.ASM294v2.21.dna.toplevel.fa: is the fasta genome file.
 - b) Schizosaccharomyces_pombe: is the key name of the new index.
4. Six different files will be generated, these files constitute the index. Create a new directory in the dataset experiment directory with the name genome and move these six files and the fasta genome file to the genome subdirectory to install it. PLEASE NOTE that the six files are required by tophat2.

5.3. Downloading the Dataset and Converting SRA Samples into FASTQ

As we have said in past sections, the complete dataset can be found here: <http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE42983>. All samples are stored in the SRA file format (Sequence Read Archive) files are a common format used by the NCBI, EBI, and others for storing reads and read alignments. We need to convert these SRA files into a most manipulable file format: FASTQ. FASTQ format is a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores. To create a fastq file ready to use in our analysis using sraToolkit:

1. Move the SRA file into the sraToolkit/bin directory.

2. Use fastq-dump script to convert the SRA file into FASTQ:

```
$ ./fastq-dump SRR636633.sra
```

Where:

- a) SRR636633.sra: is the raw RNA-Seq data experiment.

Move the new FASTQ file (SRR636633.fastq in this example) into a created subdirectory with the same name of the sample (DataSet/SRR636633/)

5.4. Map Reads to a Reference Genome Using TopHat2

As we have mention, in the pipeline we will use Tophat2 to map the RNA-Seq reads into the target genome. Tophat2 tool was used instead of Tophat, mainly because Tophat version 1 had some inconsistencies with Bowtie, but also because Tophat2 have significant best performance. To map reads into the reference genome follow these steps:

1. Create a directory GeneTranscripts and copy the .gtf file.
2. Create a directory genomeIndex and copy the genome index plus the whole genome fasta file. *IMPORTANT: Use ALL index files generated with bowtie (1,2,3,4.ebwt and rev.1 – rev.2.ebwt).
3. Execute tophat (can take a considerable amount of time):

```
$ ./tophat2 -p 4 -G
GeneTranscripts/Saccharomyces_cerevisiae.R64-1-1.75.gtf -o
SRR636633 genomeIndex/Saccharomyces_cerevisiae
SRR636633/SRR636633.fastq
```

Where:

- a) -p 4: indicates tophat to create four parallel threads. Number of threads should be selected depending of the hardware resources; in this example one thread per CPU-core was launch.
- b) -G GeneTranscripts/Saccharomyces_cerevisiae.R64-1-1.75.gtf: supply TopHat with a set of gene model annotations and/or known transcripts, as a GTF 2.2 or GFF3 formatted file. If this option is provided, TopHat will first extract the transcript sequences and use Bowtie to align reads to this virtual transcriptome first. Only the reads that do not fully map to the transcriptome will then be mapped on the genome.
- c) -o SRR636633: indicates the directory in which we want to store the results, in this example we will save the mapped reads in /SRR636633/. If the directory does not exist Tophat creates it.
- d) genomeIndex/Saccharomyces_cerevisiae: is the location and key name of the genome index generated with bowtie.
- e) SRR636633/SRR636633.fastq: is the location and fastq file of the RNA-Seq experiment we want to align.

A file called accepted_hits.bam will be created containing all mapped reads. BAM file is the compressed, binary version of SAM [17], a flexible and general purpose read alignment format This file contains the information of every read that maps into the genome.

Expression Level To firstly confirm the correct alignment of the transcriptome into the genome, and secondly investigate changes of expression levels in places of interest, a visualization of the aligned data will be done through the IGB tool, which will allow us to compare the resulting levels of expression aligned in the BAM files with a genome that contains annotations of genes. As we have seen, there are six conditions, two regarding the wild type, two with the inhibition of the Snt2 protein, and other two with the inhibition of the Ecm5 protein. Each of these pairs corresponds with two additional conditions, before and after treatment with H_2O_2 . We will compare the wild type strain after treatment against Snt2 and Ecm5 deletion strains.

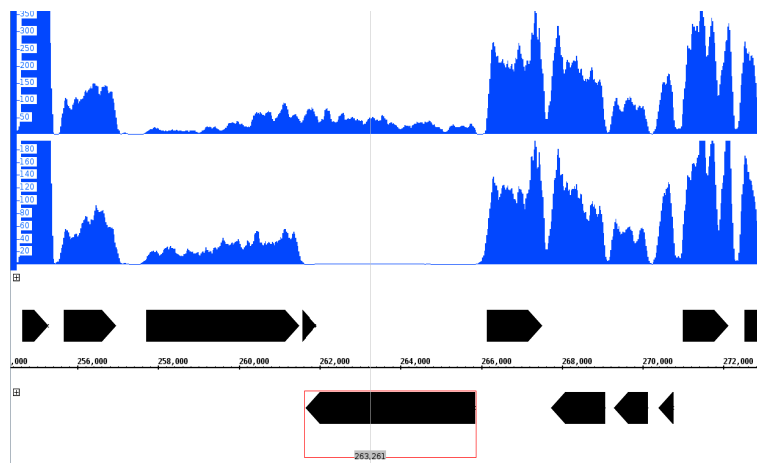


Fig. 2. WildType H_2O_2 (above) against Snt2 H_2O_2 (below)

Note the plain expression level corresponding to the Snt2 gene region in the Snt2 deletion strain after treatment (below). Thus, we can confirm that the suppression of the Snt2 gene has been achieved. Also, note the similar expression levels in the surrounding genes. As we are going to see in the next image, Ecm5 gene region in the Ecm5 deletion strain (above) have an almost complete lack of reads.

5.5. Estimating Abundance and Gene Expression With Cufflinks

Once samples have been mapped, we need to calculate abundances and gene expression. Abundance means an estimation of the number of reads at a single position; this measure can help to find significant positions within the genome. Abundance is important because it allows us to compare different experiments or samples; we can target a specific region of the transcriptome, and study the effects and the changes of the level of expression between samples. To estimate abundance and gene expression using Cufflinks follow these steps:

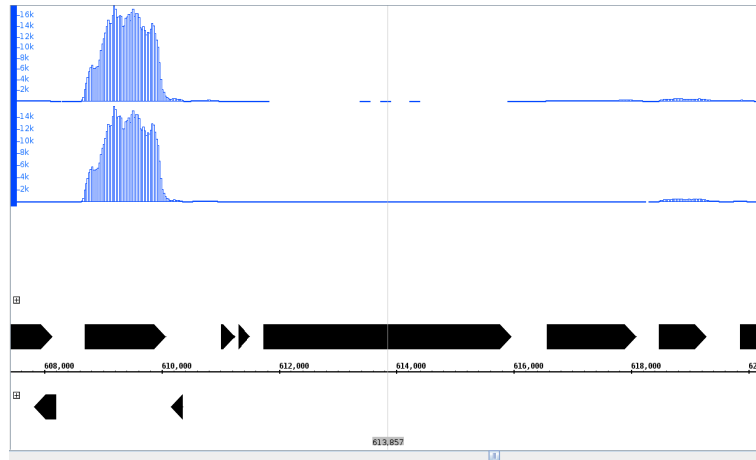


Fig. 3. Ecm5 H_2O_2 (above) against WildType H_2O_2 (below)

1. Execute cufflinks in the command line:

```
$ ./cufflinks -p 4 -G
GeneTranscripts/Saccharomyces_cerevisiae.R64-1-1.75.gtf -o
SRR636633CUFF SRR636633/accepted_hits.bam
```

Where:

- a) `-p 4`: is the number of threads that cufflinks will launch. In this example, one thread was launched per CPU-core (4).
- b) `-G GeneTranscripts/Saccharomyces_cerevisiae.R64-1-1.75.gtf`: tells Cufflinks to use the supplied gene reference annotation file to estimate isoform expression. It will not assemble novel transcripts, and the program will ignore alignments not structurally compatible with any reference transcript.
- c) `-o SRR636633CUFF`: indicates the subdirectory (in this case SRR636633CUFF) where cufflinks results will be stored. It is recommended that the subdirectory in which results from tophat are stored is different from the one that cufflinks stores its results.
- d) `SRR636633/accepted_hits.bam`: is the subdirectory and the name of the file containing the aligned reads.

A transcripts.gtf file is generated; this GTF file contains Cufflinks' assembled isoforms. We will use this file as input for the cuffdiff tool. Notably, two other files are generated:

- Genes.fpkm_tracking which contains the estimated gene-level expression values in the generic FPKM Tracking Format. We will use this file to obtain the estimated abundance of gene expression in each sample.
- isoforms.fpkm_tracking: this file contains the estimated isoform-level expression values in the generic FPKM Tracking Format.

5.6. Performing Differential Expression

To obtain a measure of how different are the results from two different conditions, we will assess differential expression comparing the expression level of each gene in two different strains. To obtain differential expression we will use cuffdiff tool [18], but firstly we need to merge all the transcripts generated by cufflinks, which will constitute the input of cuffdiff using Cuffmerge (a cufflinks tool). To do it we have to create a .txt manifest file, in which we specify the path to all the .gtf files we want to merge (corresponding to the samples we want to compare). In this example a differential expression test was done between the next conditions:

- SRR636633/4 WildType 0hours before treatment (two replicates).
- SRR636636/7 WildType 30min+ H_2O_2 after treatment (two replicates).

To perform Differential Expression with cuffdiff follow these steps:

1. Create an Assemblies.txt manifest file containing the path to all transcripts files (generated by cufflinks) where the samples are stored:

```
Assemblies.txt:
SRR636633CUFF/transcripts.gtf
SRR636634CUFF/transcripts.gtf
SRR636636CUFF/transcripts.gtf
SRR636637CUFF/transcripts.gtf
```

2. Run cuffmerge tool in the command line to merge all gene transcript files:

```
$ ./cuffmerge -g GeneTranscripts/Saccharomyces_cerevisiae.R64-1-1.75.gtf -s genomeIndex/Saccharomyces_cerevisiae.fa -p 2 wildTypeDiff/assemblies.txt
```

Where:

- -g GeneTranscripts/Saccharomyces_cerevisiae.R64-1-1.75.gtf: an optional reference annotation GTF. The input assemblies are merged together with the reference GTF and included in the final output.
 - -s genomeIndex/Saccharomyces_cerevisiae.fa: the genomic DNA sequences for the reference. The merge script will pass this option to cuffcompare, which will use the sequences to assist in classifying transfrags and excluding artifacts (e.g. repeats).
 - -p 2: number of parallel threads.
 - wildTypeDiff/assemblies.txt: directory and manifest file name containing the path to all transcripts files where the samples are stored.
3. Run cuffdiff on the command line to perform differential expression analysis between two conditions with 2 replicates each, indicating the path to the gtf merged file created by cuffmerge tool:

```
$ ./cuffdiff -o wildTypeDiff/diff_out -p 4 -L WT,WT30 -b genomeIndex/Saccharomyces_cerevisiae.fa -u wildTypeDiff/merged_asm/merged.gtf /SRR636633/accepted_hits.bam,/SRR636634/accepted_hits.bam /SRR636636/accepted_hits.bam,/SRR636637/accepted_hits.bam
```

Where:

- -o wildTypeDiff/diff_out: indicates the subdirectory (in this case ../wild-TypeDiff/diff_out) where cuffdiff results will be stored.
- -p 4: is the number of parallel threads that cuffdiff will launch. In this example, one thread was launched per CPU-core [4].
- -L WT,WT30: indicates the labels of the two conditions being compared, in this case WT for Wild Type strain and WT30 for Wild Type strain after 30 minutes of treatment with H_2O_2 .
- -b genomeIndex/Saccharomyces_cerevisiae.fa -u wildTypeDiff/merged_asm/merged.gtf: is the merged gene reference annotation file obtained with cuffmerge.
- /SRR636633/accepted_hits.bam, /SRR636634/accepted_hits.bam /SRR636636/accepted_hits.bam, /SRR636637/accepted_hits.bam: Are the SAM files of aligned RNA-Seq reads by cufflinks. If more than two are provided, Cuffdiff tests for differential expression and regulation between all pairs of samples. If more than one replicate for a sample exists, supply the SAM files for the sample as a single comma-separated list. It is not necessary to have the same number of replicates for each sample. In this example SRR636633,SRR636634 are two replicates of Wild Type before treatment, and SRR636636,SRR636637 are the two replicates of the Wild Type 30 minutes after treatment with H_2O_2 .

Cuffdiff generates a wide range of files as output. We will be using two files:

- *genes.fpkm_tracking*: Cuffdiff calculates the FPKM of each gene in each sample. The gene FPKMs are computed by summing the FPKMs of transcripts in each gene group.
- *gene_exp.diff*: This tab delimited file lists the results of differential expression testing between samples for genes. The values are the difference between summed FPKM of transcripts sharing each *gene_id*.

5.7. Differential Expression Results

In this experiment, differential expression tests were carried between the next datasets [16]:

- Between WILD TYPE before treatment and WILD TYPE 30' after treatment with H_2O_2 .
- Between Snt2 before treatment and WILD TYPE before treatment.
- Between Ecm5 before treatment and WILD TYPE before treatment.
- Between Snt2(H_2O_2) and WILD TYPE 30' after treatment.
- Between Ecm5(H_2O_2) and WILD TYPE 30' after Treatment.

The goal of this is, firstly, to discover the number of genes that vary expression levels with a minimum significance between the Wild Type strains before and after treatment. This will tell us which genes are triggered or suppressed in a wild type strain on an oxidative stress condition with H_2O_2 . Then we compare all the rest conditions: the end of this is to discover changes in expression levels

on an oxidative stress environment in strains that had suffered the deletion of a single gene (Snt2 or Ecm5). Differential expression analysis between Wild Type before and after treatment conditions showed a great change in gene expression levels (Illustration 4 Wild Type against Wild Type(H_2O_2) significance Volcano Plot). The analysis identified 2973 genes that significantly changed expression in response to H_2O_2 (positive value means that the gene has increased its value between conditions, thus we say that the gene is over expressed). In comparison,

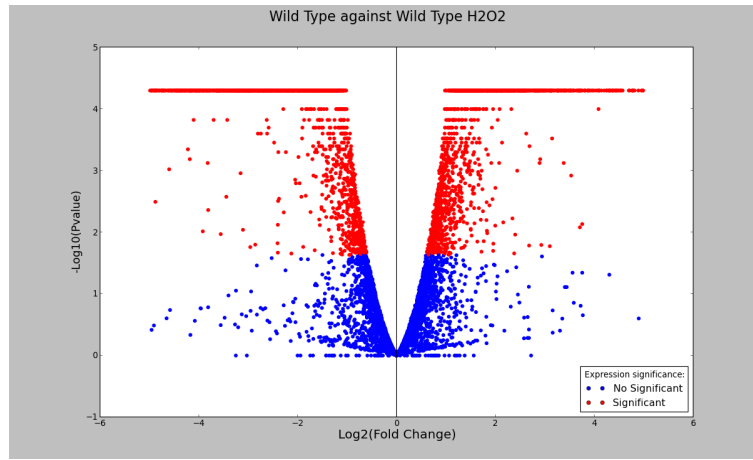


Fig. 4. Wild Type against Wild Type(H_2O_2) significance Volcano Plot

differential expression analysis between Ecm5 deletion strain and Wild Type before (Illustration not included, for further information consult the entire study attached) and after (See Illustration 5 Ecm5 against Wild Type(after H_2O_2 treatment) significance Volcano Plot) treatment revealed no significant genes. This suggests that Ecm5 has a secondary role in gene expression levels [16]. In contrast, the analysis of Snt2 between Wild Type before treatment showed 54 significant changes in expression levels, and more striking, differential expression analysis between Snt2 deletion strain and Wild Type strain after treatment with H_2O_2 (See Illustration 6 Snt2 against Wild Type(after H_2O_2 treatment) significance Volcano Plot) showed 357 genes with significant change in expression level. These expression results suggest the role of Snt2 gene to both promote and repress gene expression, depending on the target. These 357 genes (Table 1 Significant Gene Names) obtained from Snt2 strain differential expression test against Wild Type with significant change in expression level will be used for further analysis: since Snt2 is believed to be involved in the regulation of gene expression we will try to find similar expression patterns between the 357 significant expressed genes using data mining and statistical techniques.

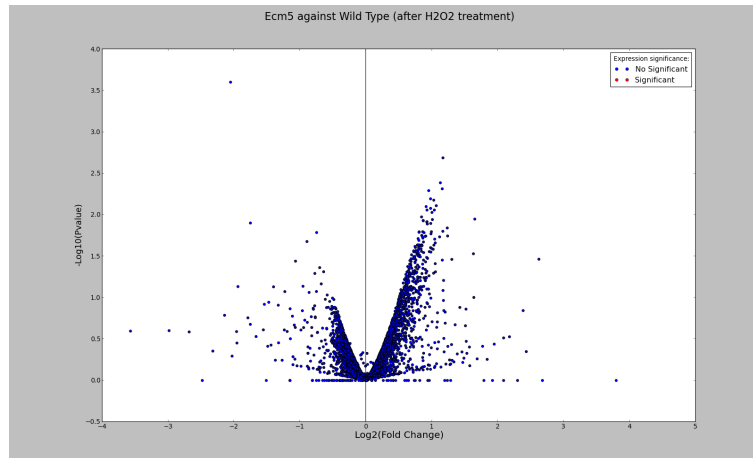


Fig. 5. Ecm5 against Wild Type(after H_2O_2 treatment) significance Volcano Plot

The Cell Wall Mannoprotein Genes FIT1 and DAN1 Special cases of study are both cell wall mannoprotein genes FIT1 and DAN1. Snt2 and Ecm5 are believed to be related with cell wall regulation functions [16], to prove it, the differential expression results of DAN1 and FIT1 genes were examined in conditions Snt2 after H_2O_2 treatment and Ecm5 after H_2O_2 treatment, both compared against Wild Type after H_2O_2 treatment. The differential expression tests showed opposite expression changes of FIT1 and DAN1 genes in the two strains, in Snt2 condition (red) both genes are sub-expressed. These results link Snt2 and Ecm5 function to cell wall regulation and are consistent with Snt2 and Ecm5 having opposing functions [16].

Oxidative Stress Response (OSR) and Cytochrome c Pathway Target Genes As it has been said, Snt2 is believed to be involved in the oxidative stress response, that is, it is believed that Snt2 has a role in regulating genes that participate in the oxidative stress response. A group of seven significant target genes (SUE1, HSP12, CYC3, CYC1, CIN5, CTT1, HSP30) that participate in either the oxidative stress response or the cytochrome c pathway were examined, and showed that genes were up or down regulated in the snt2 strain after H_2O_2 treatment.

Many of the Snt2 and Ecm5 targets involved in the oxidative stress response or incytochrome c regulation had altered expression levels in the Snt2 strain after H_2O_2 treatment [16], these results show that H_2O_2 treatment prompts Snt2 to target and regulate stress response genes.

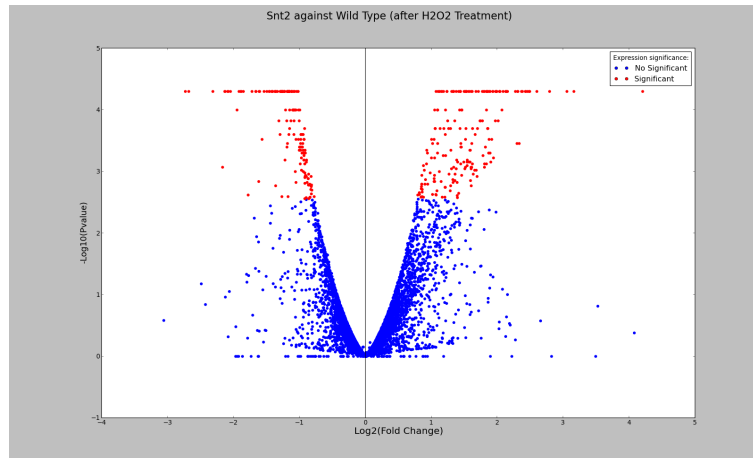


Fig. 6. Snt2 against Wild Type(after H_2O_2 treatment) significance Volcano Plot

6. Study Over Significant Expressed Genes

The group of interest is formed by 357 genes (see Table 1 Significant Gene Names obtained from Snt2 strain differential expression test). In order to discover which functions links these genes with each other, find patterns among genes and conditions, and in general try to abstract this large amount of data to infer valuable information, a data mining study will be carried on the data. To perform the following analysis, we need to build a matrix containing these 357 genes with the original RPKM values of the reads obtained by cufflinks in all the six conditions.

6.1. K-Means Clustering

K-Means algorithm [19] was applied using Expander tool [20], and as a distance metric the Pearson's correlation distance was used. The starting points of the clusters were selected at random. Four distinctive clusters were found.

Cluster 1 The cluster is formed by 54 genes. This cluster revealed a significant over-expression profile in condition Snt2 after treatment (red), and furthermore conditions Wild Type after H_2O_2 treatment and Ecm5 after H_2O_2 treatment have a slightly higher average value in contrast with those of the three before treatment conditions (the three top conditions in the y-axis on the left of the image).

Cluster 2 The cluster is joined by 103 genes. In this group, an over-expression pattern was found in conditions Wild Type after H_2O_2 treatment and Ecm5 after H_2O_2 treatment. Contrary to the previous cluster, Snt2 condition showed

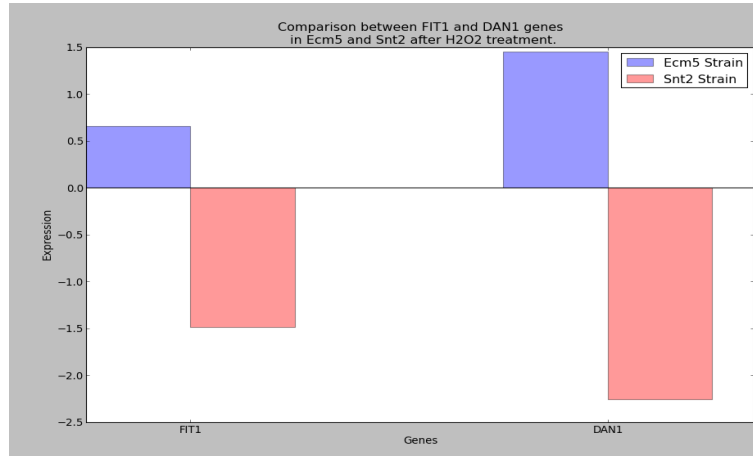


Fig. 7. FIT1 and DAN1 genes expression on Ecm5 and Snt2 Strains

a sub-expressed profile. In fact, Snt2 after H_2O_2 treatment condition have the same average value of that of the other three before treatment condition, as we can see in the diagram of the expression profile of cluster 2.

Cluster 3 The third cluster is formed by 36 significant genes. Strikingly, the expression pattern found in this cluster is completely opposite to the expression of cluster 2, although we have to mention that the third cluster is the smallest group in the clustering analysis. Snt2 after H_2O_2 treatment and the three conditions (WildType, Ecm5 and Snt2) before treatment have a similar over-expressed profile, whereas WildType after H_2O_2 treatment and Ecm5 after H_2O_2 treatment have a clear sub-expressed profile (green). Graphical representation of the cluster's expression profile showed a similar average value of the over-expressed conditions, and a clearly drop value of gene expression in both WildType after H_2O_2 treatment and Ecm5 after H_2O_2 treatment. The founding of this distinctive cluster is quite interesting, because is not mentioned in the original study.

Cluster 4 140 genes with a similar pattern form this cluster. In this case, all conditions after H_2O_2 treatment (Snt2, WildType and Ecm5) are generally sub-expressed (green). All before conditions (Snt2, WildType and Ecm5) are this time over-expressed (red). As we will see in the next chapter this fourth cluster is, at the same time, formed by other 3 sub-clusters.

6.2. Hierarchical Clustering

To consolidate the results obtained by using K-Means algorithm, a similar analysis on the same dataset was carried. Expander hierarchical clustering was

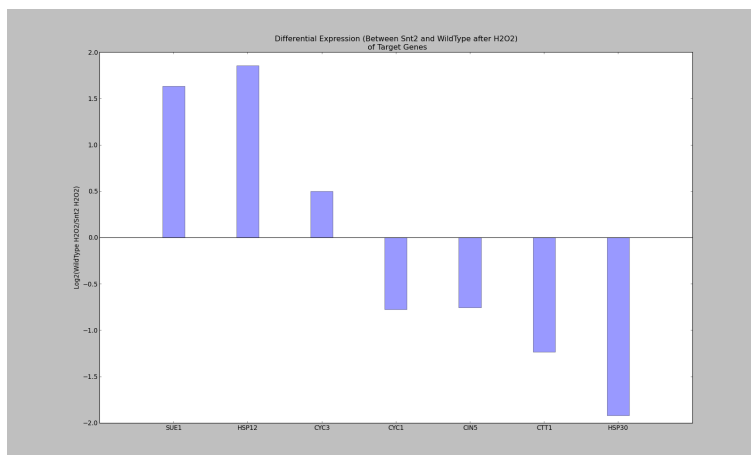


Fig. 8. Log2 ratios of expression levels of genes: SUE1, HSP12, CYC3, CYC1, CIN5, CTT1, HSP30, in the Snt2 strain after treatment relative to WildType levels after treatment

used, and the chosen distance metric was the Euclidean distance. The hierarchical clustering threw a similar set of results with four major clusters, furthermore cluster 4 (similar that the one of the previous section) joined by 140 genes showed to be formed by 3 sub-clusters.

Cluster 3 Similarly to K-Means results, hierarchical clustering threw strong evidences of a distinctive group with a similar gene expression pattern. Cluster is formed by 33 genes which all of them are in the list of 36 genes of the cluster 3 generated by K-Means algorithm. As we said in the last section, this distinctive cluster is quite interesting because is not mentioned in the original study.

Cluster 4 Other interesting group is the cluster 4, the largest of all with 140 genes. In the original paper [16] four clusters are distinguished, but the cluster corresponding with our cluster 4 is formed by three clusters, using hierarchical clustering we identify the three clusters of the original experiment. Hierarchical clustering obtained the original four clusters, including this. All these results corroborate the original cited article.

6.3. Gene Ontology and Functional Analysis

Using David [21] (Database for Annotation, Visualization and Integrated Discovery) whose primary purpose is to provide functional interpretation of large lists of genes obtained from genomic studies, we will try to discover shared functionalities between genes in the same cluster.

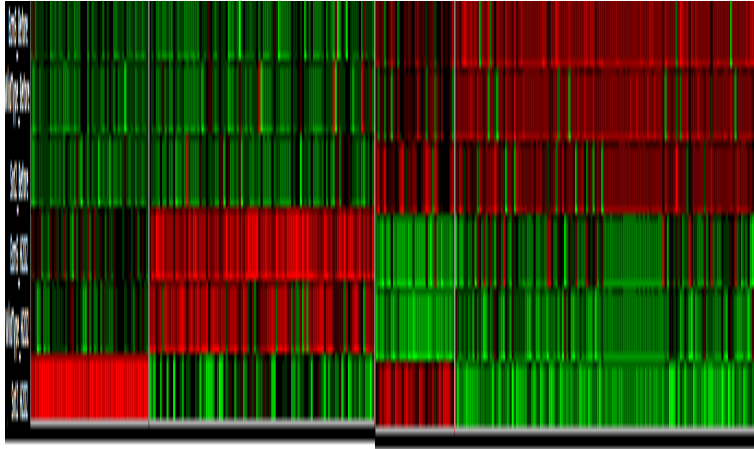


Fig. 9. Gene expression matrix with the 357 significant genes, and 6 conditions (from top to bottom: Ecm5_before, WildType_before, Snt2_before, Ecm5_H₂O₂, WildType_H₂O₂, Snt2_H₂O₂). Red: Over-Expressed, Green: Sub-Expressed

Classification of Cluster 1 Functional analysis found that 6 of the 54 genes are related with the term ribosome biogenesis. Besides, 6 genes were related with the term signal, and 5 genes were related to the term rRNA processing. Gene ontology classification showed that 10 out of 54 genes are related with response to temperature stimulus, furthermore 10 genes are related with response to abiotic stimulus. Notably, 9 genes were linked to cellular response to stress. Worth noting, 7 genes were related to oxidation reduction. These results again link Snt2 with ESR (environment stress response) genes [16].

Classification of Cluster 2 Functional analysis linked 54 genes out of 103 with the term phosphoprotein (a protein which is posttranslationally modified by the attachment of either a single phosphate group, or of a complex molecule, such as 5'-phospho-DNA, through a phosphate group). Also, 20 genes were related with the term transferase. Notably, 16 were related with the term oxidoreductase. Gene ontology classification showed that 18 of the 103 genes are linked with oxidation reduction (which again links Snt2 and Ecm5 with oxidation stress response). Also, 17 genes were related with nitrogen compound biosynthetic process.

Classification of Cluster 3 Functional analysis found that 16 of 36 genes are related with the term transmembrane protein. In addition, 15 are linked with the term membrane and 14 with the term transmembrane. Gene ontology classification threw that 15 genes of 36 are related with integral to membrane functions, and 15 with intrinsic to membrane. Worth noting, 9 genes are linked

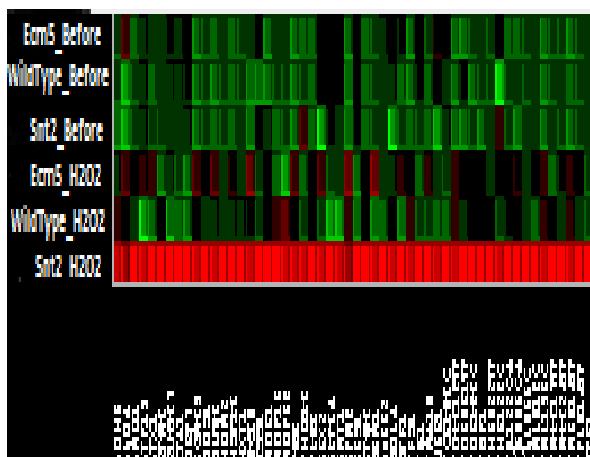


Fig. 10. K-Means Cluster 1 Expression Matrix

with plasma membrane. These results again link Snt2 (and in particular the group of genes in this cluster) with membrane functionalities.

Classification of Cluster 4 Functional analysis showed that 100 of 136 genes are related with the term phosphoprotein. Moreover, 53 genes were related with the term nucleus, and 46 with the term cytoplasm. Gene ontology classification showed that 79 of 136 genes are involved in intracellular non-membrane-bounded organelle and non-membrane-bounded organelle. Also 65 genes are involved in ribonucleoprotein complex. Notably 49 genes are related to membrane-enclosed lumen.

7. Conclusions

RNA-Seq has proved to be a high valuable technology: since its initial application it has developed the way of making transcriptome profiling. Advances in recent years have provided an even more complete characterization of RNA transcripts, throwing a far more precise measurement of levels of transcripts and their isoforms than other methods. In contrast, the huge number and wide diversity of file formats, the data and knowledge dispersion across biological databases and platforms, the variety of tools and not-interoperable software, the different (and sometimes incompatible) measurement rules for abundance, counting and gene expression, and the diversity of algorithmic approaches to assess results makes genome sequencing and transcriptome profiling a challenging, and worst of all, chaotic task when processing RNA-Seq data output.

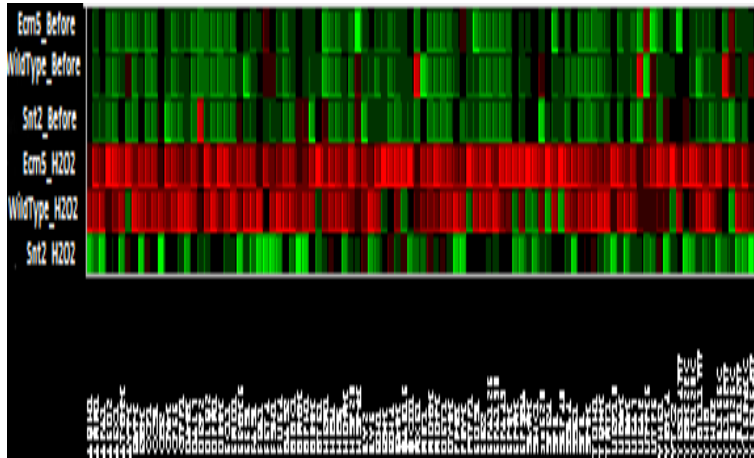


Fig. 11. K-Means Cluster 2 Expression Matrix

7.1. Experiment Biological Conclusions

In *Saccharomyces cerevisiae* (a species of yeast) the response to stress involves regulation of a common set of genes known as ESR genes (Environmental Stress Response genes). Snt2 and Ecm5 genes are believed to be related in some way to the stress response (and in particular to the oxidative stress response) of the cell. The original experiment showed the physical association between Snt2 and Ecm5 genes (16), and have integrated RNA-seq analyses to uncover a role for Snt2 in regulating genes in response to oxidative stress. Also worth noting, the original paper showed evidences of Snt2 and Ecm5 being involved in cell wall regulation functions. In this study we corroborate the original study conclusions, through the examination of localization changes of either Snt2 or Ecm5 in response to stress using different computational methods, in particular, a group of significant genes related to Snt2 gene, which showed distinctive patterns among its expression values was isolated. Using this group of significant genes as a sub-dataset we carry a pattern recognition study using data mining techniques, like K-Means clustering and Hierarchical clustering. The obtained results were totally consistent with the original study; furthermore a distinctive small cluster that was not in the original study was discovered, thus this cluster was later demonstrated to be related with transmembrane and membrane functions using an ontology and functionality study. Also, physical association between Snt2 and Ecm5 genes and the link of Snt2 and Ecm5 with cell wall proteins was shown using visualization techniques.

7.2. Work Conclusions

The main goal of this paper was to show a complete guide over a RNA-Seq study, from the original raw RNA-Seq data to the data mining and differential

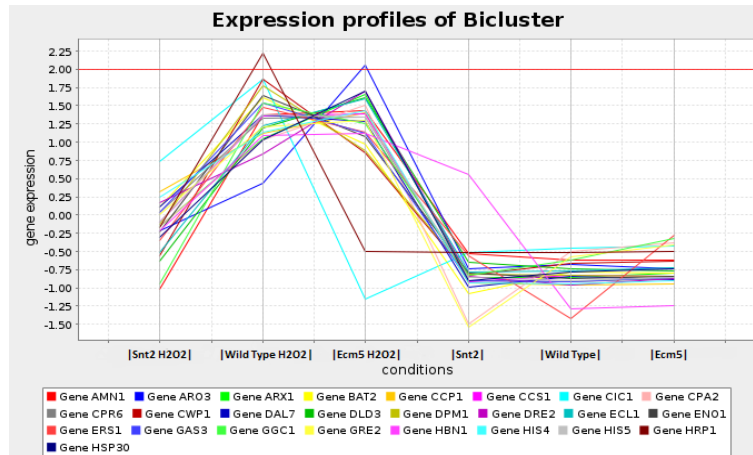


Fig. 12. Expression Profile of Cluster 2 Representing the First 25 genes

expression results. Thus, a complete description of a typical RNA-Seq pipeline has been shown, and furthermore, some post-processing techniques have been applied to the obtained data, which is not always present in these kinds of experiments. What we have seen here is the enormous possibilities of some computing techniques applied to a massive set of biological data, and it has become evident that huge possibilities arise with these new technologies: there is lot of work yet to be done in this particular field. Another important and personal key is that this experiment has allowed me to familiarize myself with a wide variety of tools and programming languages involved in the field of bioinformatics, which may lead me to a professional career in this sector. Another remarkable subject is the study of some data mining techniques, which nowadays are decisive in all kind of fields and business due to the large amount of data that is being generated today.

And last but not least, this study has allowed me to take the first steps into a real and serious scientific work.

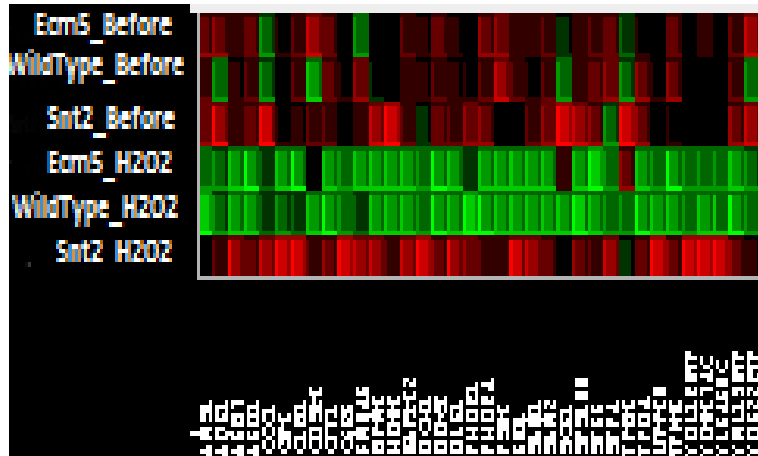


Fig. 13. K-Means Cluster 3 Expression Matrix

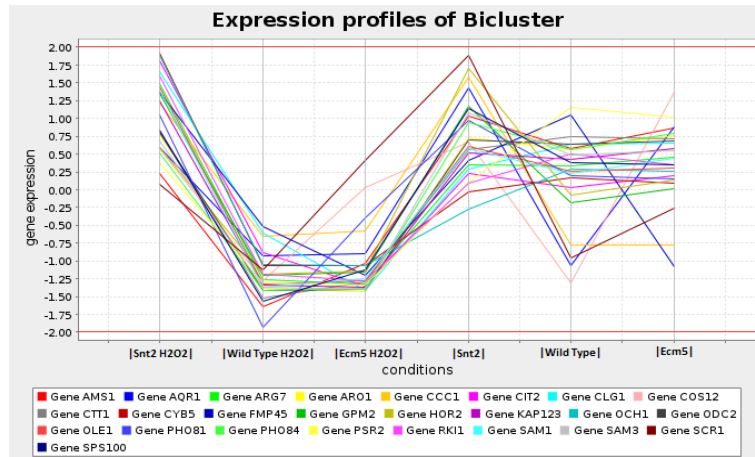


Fig. 14. Expression Profile of Cluster 3 Representing the First 25 genes

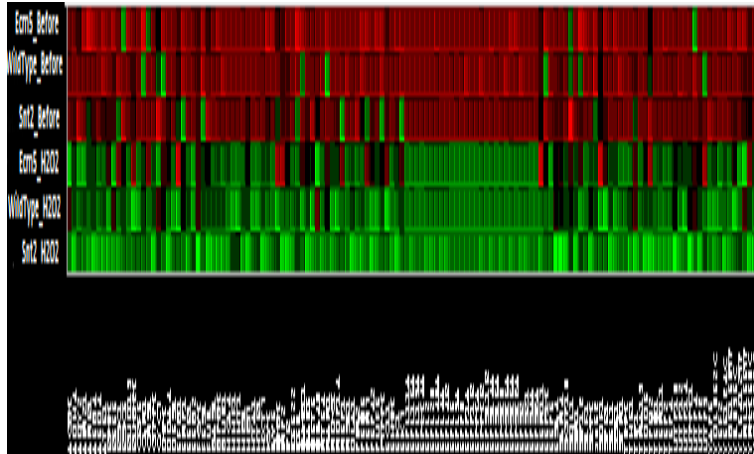


Fig. 15. K-Means Cluster 4 Expression Matrix

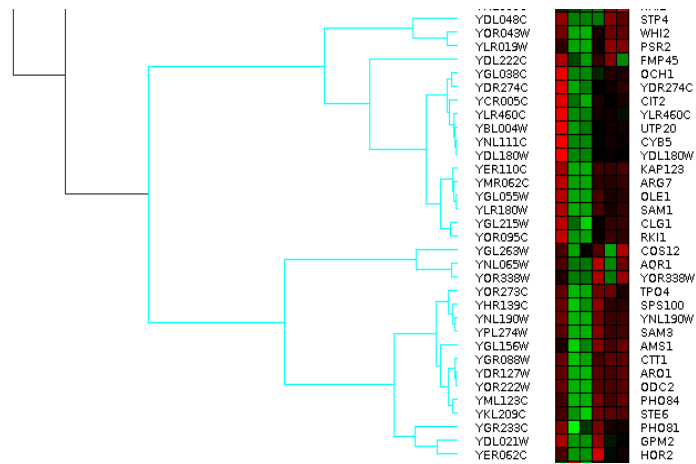


Fig. 16. Hierarchical Clustering 3

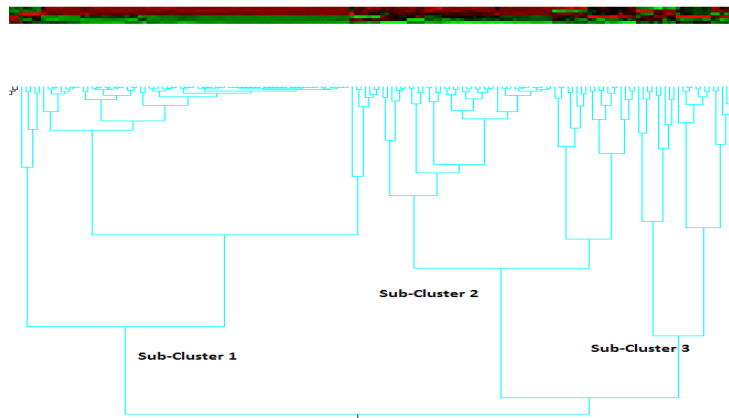


Fig. 17. Hierarchical Clustering Cluster 4

Bibliography

- [1] J. A. M. Wang and Zhong, "Next-generation transcriptome," *Nature Reviews*, vol. 12, 2011.
- [2] Z. Wang, M. Gerstein, and M. Snyder, "Rna-seq: a revolutionary tool for transcriptomics," *Nature Reviews*, vol. 10, 2009.
- [3] K.-O. Mutz, A. Heilkenbrinker, M. Lonne, J.-G. Walter, and F. Stahl, "Transcriptome analysis using next-generation sequencing," *Current Opinion in Biotechnology*, vol. 24, 2003.
- [4] F. O. Milos and P. M., "Rna sequencing: advances, challenges and opportunities," *Nature Reviews*, vol. 12, 2011.
- [5] M. L. Metzker, "Sequencing technologies — the next generation," *Nature Rev. Genet.*, vol. 11, pp. 31–46, 2010.
- [6] B. Homer, N. Merriman and S. Nelson, "Bfast: an alignment tool for large scale genome resequencing," *PLoS ONE*, vol. 4, 2009.
- [7] Z. Smith, A. D. Xuan and M. Zhang, "Using quality scores and longer reads improves accuracy of solexa read mapping," *BMC Bioinformatics*, vol. 9, 2008.
- [8] B. Langmead, C. Trapnell, M. Pop, and S. Salzberg, "Bowtie: Ultrafast and memory-efficient alignment of short dna sequences to the human genome," *Genome Biology*, vol. 10, 2009.
- [9] P. Ferragina and G. Manzini, "An experimental study of a compressed index," *Inf. Sci.*, vol. 135, 2001.
- [10] C. Trapnell, L. Pachter, and S. Salzberg, "Tophat: discovering splice junctions with rna-seq," *Bioinformatics*, vol. 25, pp. 1105–1111, 2009.
- [11] Y. Katz, E. T. Wang, E. M. Airoidi, and C. B. Burge, "Analysis and design of rna sequencing experiments for identifying isoform regulation," *s.l. : Nature Methods*, vol. 7, 2010.
- [12] C. Trapnell, A. Roberts, L. Goff, G. Pertea, D. Kim, D. R. Kelle, H. Pimentel, S. L. Salzberg, J. L. Rinn, and L. Pachter, "Differential gene and transcript expression analysis of rna-seq experiments with tophat and cufflinks," *s.l. : Nature Methods*, vol. 7, 2012.
- [13] C. e. a. Trapnell, "Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation," *Nature Biotech.*, vol. 28, 2010. The Cufflinks algorithm is introduced in this paper, which was one of the first reference-based assemblers that effectively tackled the assembly of alternative isoforms using NGS data.
- [14] L. H. *, H. B. *, W. A., F. T., R. J., H. N., M. G., A. G., D. R., and . G. P. D. P. Subgroup, "The sequence alignment/map (sam) format and samtools," *s.l. : Bioinformatics*, vol. 25, 2009.
- [15] N. JW, H. GA, B. S. Jr, R. A, and L. AE., "The integrated genome browser: free software for distribution and exploration of genome-scale datasets," *s.l. : Bioinformatics*, vol. 25, 2009.

- [16] L. A. Baker, B. M. Ueberheide, S. Dewell, B. T. Chait, D. Zheng, and C. D. Allis, "The yeast *snt2* protein coordinates the transcriptional response to hydrogen peroxide-mediated oxidative stress," *s.l. : Molecular and Cellular Biology*, vol. 33, 2013.
- [17] H. e. a. Li, "The sequence alignment/map format and samtools," *s.l. : Bioinformatics*, vol. 25, 2009.
- [18] M. Garber, M. G. Grabher, M. Guttman, and C. Trapnell., "Computational methods for transcriptome annotation and quantification using rna-seq," *Nature Methods*, vol. 8, 2011.
- [19] R. Shamir, T. J. Sharan, R. Algorithmic, Y. X. T. Smith, and M. Z. editors, "Approaches to clustering gene expression data. in current topics in computational biology," *s.l. : MIT Press*, 2002.
- [20] R. Sharan, A. Maron-Katz, and R. Shamir, "Click and expander: A system for clustering and visualizing gene expression data," *s.l. : Bioinformatics*, vol. 19, 2003.
- [21] G. D. Jr., B. T. Sherman, D. A. Hosack, J. Yang, M. W. Baseler, H. C. Lane, and R. A. Lempicki, "David: Database for annotation, visualization, and integrated discovery," *s.l. : Genome Biology*, vol. 4, 2003.

Sistema de predicción de edad en rostros

Alfonso González Briones¹, Juan Manuel Corchado Rodríguez¹ y Juan Francisco de Paz Santana¹

¹Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España
{alfonsogb, corchado, fcofds}@usal.es

Resumen En este trabajo se realiza el diseño de un sistema para estimar la edad de los rostros identificados en una imagen, utilizando cámaras que nos permitan obtener imágenes en tiempo real. El objetivo es clasificar las imágenes faciales en diversos tramos de edad. Para ello, se presenta el estado del arte en el área de Reconocimiento Facial, Estimación de la edad a través de rostros humanos, algoritmos Eigenfaces y Fisherfaces. Como resultado se ha obtenido un sistema, entrenado sobre una máquina de aprendizaje con la base de datos FERET, está basado en el clasificador Fisherfaces, complementado con el filtro no lineal Bilateral para reducir el ruido y preservar la dirección de la imagen original.

Keywords: Reconocimiento facial, Estimación Automática de edad, Fisherfaces, Preprocesado de imágenes, FERET

1. Introducción

El rostro humano encarna una rica cantidad de información utilizable en muchas aplicaciones interesantes, una de las más fascinantes de clasificación es la clasificación automática de la edad. El problema ha inspirado a muchos investigadores que han conducido diversos conjunto de soluciones, pero una observación significativa entre las propuestas es que su viabilidad se ha evaluado principalmente en entornos controlados. En este sentido, es necesario hacer más para demostrar que la clasificación por edades vale la pena en escenarios generales y realistas, es decir, en entornos adquiridos sin restricciones de condiciones.

Este proceso de determinación de la edad, otorga a las máquinas la capacidad de tomar decisiones en base al rango de edad estimado. Estas decisiones tiene cabida en una variedad de aplicaciones que van desde el control de acceso, la interacción hombre-máquina, la identificación de personas, la minería de datos y la organización. Las aplicaciones típicas para cada categoría mencionada anteriormente incluyen [1]:

- **Control de acceso basada en la edad:** Aplicar una estimación automática de la edad facial, de forma precisa y no invasiva, de la persona que desea acceder a un dominio físico o virtual específico.

- **Edad adaptativa Interacción Hombre-Máquina (HCI):** Una estimación automática de la edad puede ser utilizada para determinar la edad de un usuario de la computadora / maquina y automáticamente ajustar la interfaz de usuario con el fin de satisfacer las necesidades de su grupo de edad.
- **Identificación de Personas de edad invariante:** Los algoritmos de progresión de edad suelen requerir información relacionada con la edad actual de una persona, por lo tanto, un sistema preciso de estimación de la edad facial puede desempeñar un papel clave en el desarrollo de sistemas automáticos de progresión de edad, de manera que apoye en la verificación de la identidad en edad invariante.
- **Minería de datos y organización:** Los sistemas de estimación de la edad se pueden utilizar para la recuperación de información basada en la edad y la clasificación de imágenes de rostros que permitan de esta manera la clasificación automática y recuperación de imágenes de los álbumes web de fotos y de Internet.

Muchos métodos de representación de la imagen han sido estudiadas como modelos antropométricos, modelos de apariencia activa (AAM) o patrones de envejecimiento subespacial. Se puede encontrar una extensa revisión de estos métodos de representación edad en [2] y [3]. En cuanto a los esquemas de clasificación de la edad, los métodos existentes se basan en pura clasificación, o en análisis de regresión. Tal vez, entre los estudios pioneros en clasificación de edades sean los propuestos por Kwon y Lobo [4], Lanitis et al. [5], y Guo et al. [6]. Aunque con relativo éxito en algunos escenarios (por ejemplo, de alta calidad e imágenes libre de oclusión, imágenes libre de fondo y expresión facial neutra), la mayoría de los métodos existentes tienden a sufrir ajustes no controlados en la configuración como se indica en [7].

A diferencia de los trabajos mencionados anteriormente, en este trabajo se propone un nuevo método para la clasificación automática de la edad sin restricción de condiciones. Centrando nuestra investigación hacia el desarrollo de un esquema de reconocimiento de patrones que no dependa en exceso de la geometría y de cálculos como plantillas deformables. El uso de Fisherfaces parece ser un método adecuado para ser utilizado en el reconocimiento de caras, debido a su capacidad, simplicidad, velocidad y aprendizaje [8].

2. Estado del arte

Desde que apareciera el primer estudio relacionado con la clasificación y/o estimación de la edad, a través de la información facial, propuesto por Kwon and da Vitoria Lobo [4] han surgido muchos estudios cada uno con diversas técnicas y diferentes formas de tratar los datos de entrada por parte de los métodos aplicados.

Aunque se han realizado estudios para tratar esta temática con redes neuronales Dehshibi and Bastanfard [9] la dirección que están tomando las investigaciones se centra en la aplicación de algoritmos matemáticos y/o estadísticos

al conjunto de datos. Los estudios de estimación de la edad ha continuado de cierta forma los relacionados con el reconocimiento facial, algunos algoritmos de clasificación como Eigenfaces o Fisherfaces pronto se aplicaron en este área.

Estos algoritmos para poder decidir entre las distintas franjas de edad, es necesario aprendan las características de cada una de ellas, y en esta tarea el método Eigenfaces basado en el análisis de componentes principales, no es un método totalmente adecuado al ser un modelo estadístico sin supervisión. Sin embargo, Fisherfaces al producir una proyección lineal de la clase específica, es mucho más adecuado para el cometido, como demostró Seo [10].

No obstante, los trabajos no solo se están enfocando en aplicar, mejorar, combinar algoritmos o desarrollar otros nuevos sino que rápidamente se centraron en la extracción de información de los rostros como planteó Choi et al. [11], siendo de los primeros sistemas más completos que incluían la combinación de características globales y locales.

Otras menos, como la investigación enunciada por Jana et al. [12] se encaminan por los cauces del cálculo de proporciones biometricas y los ángulos que se forman por los distintos elementos: ojos, nariz y boca. No obstante este tipo de investigaciones se suelen complementar con la extracción de características puesto que solo tiene un buen rendimiento en edades tempranas, al ser la etapa en la que mayores cambios se producen en las medidas del rostro [13].

Por ello durante el estudio hemos analizado 3 algoritmos (Fisherfaces, Eigenfaces, LBP) muy utilizados en la literatura, empleados para la extracción de las características faciales, reducción de la dimensionalidad o empleados para clasificar en las distintas franjas.

Para poder aplicar los algoritmos de forma correcta y obtener todavía mejores resultados, es imprescindible realizar un preprocesado de las imágenes. Es por esto, que también ha sido necesario realizar un análisis de diferentes filtros para conseguir unos datos de entrada limpios de datos innecesarios para el proceso de estimación.

2.1. Fisherfaces

Como típico método holístico, el análisis discriminante lineal (LDA) ha sido muy exitoso en la clasificación de patrones de alta dimensión, tales como reconocimiento facial desde la aparición del método Fisherfaces. Actualmente todos los sistemas de reconocimiento facial surgidos están basados en LDA.

Sin embargo cuando se aplica LDA para reconocimiento facial, siempre nos podemos encontrar con el problema del pequeño tamaño de la muestra (Small Sample Size - SSS), donde el número de muestras de entrenamiento N , es mucho menor que la dimensión de la imagen facial. Hasta la fecha, muchos investigadores han propuesto diferentes maneras de solucionar este problema. El enfoque más famoso es Fisherfaces, fue presentada por Swets [14] y Buenhumor [15] utiliza el análisis de componentes principales (PCA) para mapear los datos originales para un subespacio de baja dimensión a fin de eliminar la singularidad de S_w .

Fisherfaces ha mostrado ser mejor que Eigenfaces particularmente bajo condiciones en la que se producen variaciones en la iluminación y la expresión facial.

Esto es debido a que el principio básico de Fisherfaces en lugar de calcular un eigenvector especial para cada imagen del conjunto de entrenamiento, sólo calcula un eigenvector especial y eigenvalue para cada persona.

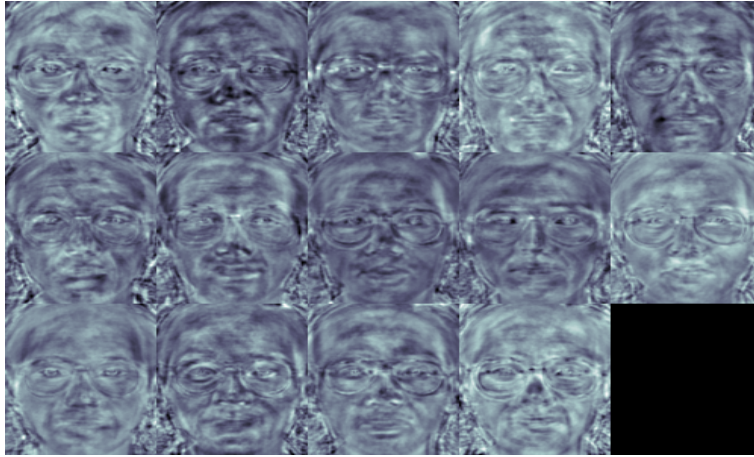


Figura 1. Fisherfaces

2.2. Filtrado

Uno de los problemas más comunes que nos encontramos a la hora de analizar imágenes obtenidas desde dispositivos de captura es el ruido de compresión, cuantización y de sensibilidad del sensor de captura.

Uno de los métodos existentes para reducir ese ruido es filtrar espacialmente la imagen (suavizado) aunque ello provoque pérdida del detalle de la imagen.

Filtro Bilateral: El filtrado Bilateral suaviza las imágenes sin alterar los bordes, por medio de una combinación no lineal de los valores cercanos de la imagen. El método es no iterativo, local y simple. Combina los niveles de grises o de color en función de su cercanía tanto geométrica y su similitud fotométrica, prefiere los valores próximos a los valores de distancia tanto en dominio como en rango. En contraste con los filtros que operan en las tres bandas de color por separado, un filtro bilateral puede hacer cumplir la métrica de percepción subyacente en el espacio de color CIE-Lab, alisar los colores y preservar los bordes de una manera que se ajusta a la percepción humana. También, en contraste con el filtrado estándar, el filtrado bilateral no produce colores fantasma a lo largo de bordes en imágenes en color, y reduce los colores fantasma que aparecen en la imagen original,[16].

2.3. Bases de datos

En la actualidad existen tan solo dos conjuntos de datos de libre acceso que tienen como objetivo apoyar la experimentación en el área del envejecimiento facial (MORPH (Ricanek and Tesafaye [17] 2006) y FG-NET (Lanitis [18] 2008)).

Sin embargo no cumple con los requisitos para ser un conjunto adecuado de datos, la base de datos MORPH contiene sólo unas pocas muestras por sujeto, mientras que la base de datos FG-NET contiene imágenes que muestran una variación significativa no relacionada con el envejecimiento.

FERET: La Base de datos FERET es la norma de facto en la evaluación de un Sistema de reconocimiento facial. Las imágenes fueron tomadas bajo condiciones semi-controladas para poder evaluar los algoritmos de reconocimiento facial. Esto contrasta con muchos de los algoritmos de la literatura, donde se presentan los resultados para imágenes de pequeñas bases de datos recogidas en condiciones altamente controladas.

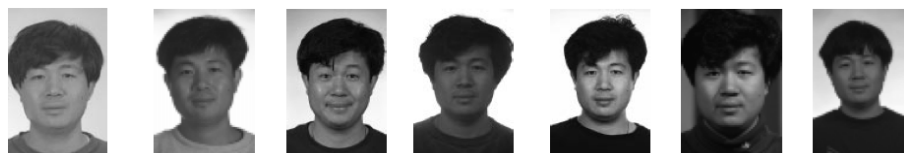


Figura 2. Imágenes de FERET de un mismo sujeto

La base de datos FERET hizo posible que los investigadores desarrollase algoritmos en una base de datos común y que se pueda informar de los resultados en la literatura del uso de esta base de datos. Los resultados reportados en la literatura tampoco proporcionaron una comparación directa entre los algoritmos, ya que cada investigador presenta los resultados utilizando diferentes supuestos, métodos de puntuación y diferentes conjuntos de imágenes. Las evaluaciones con FERET administradas de forma independiente permite una evaluación cuantitativa directa de las fortalezas y debilidades de los diferentes algoritmos.

3. Descripción del sistema propuesto

Todos los objetivos mencionados previamente los podemos englobar en uno más general: Crear un sistema que nos permita clasificar en los diferentes tramos de edad propuestos a las personas halladas por cámaras en tiempo real. Para llegar a éste se propone una arquitectura base general que se encarga de realizar todos los procesos necesarios para ello.

Este sistema, como se ve en la figura 3, necesita de otros subsistemas y herramientas de apoyo para ser capaz de llevar a cabo la labor de clasificación por tramos.

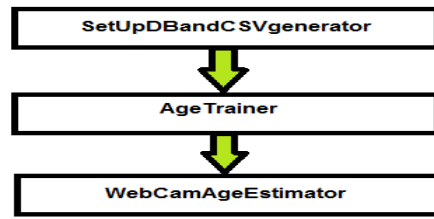


Figura 3. Estructura del sistema propuesto

3.1. SetUpDBandCSVgenerator

El primer subsistema es el encargado de preparar la base de datos y generar un CSV para el proceso de entrenamiento. Para ello se carga un Clasificador en cascada Haar entrenado para reconocer ojos humanos en una imagen.

Para preparar la base de datos de forma adecuada, y poder entrenar de la mejor manera factible, el sistema recorre cada directorio (tramo de clasificación de edad) imagen por imagen, realizando un preprocesado de las imágenes siguiendo los sucesivos pasos:

- El clasificador en cascada Haar es aplicado a la imagen, recorriendo la imagen varias veces para detectar los ojos. Una vez detectados los ojos, recojemos las coordenadas de cada ojo del rostro de la imagen.
- Comprobamos si la imagen esta en la posición correcta. Se comprueba si la coordenada 'x' del primer ojo es menor que la coordenada x del segundo ojo, la imagen esta al revés, y se gira para ponerla en la posición correcta.
- Con las coordenadas de ambos ojos, se calcula el punto medio entre ellos. Con la coordenada 'x' del punto medio de los ojos, se realiza el centrado en el eje horizontal, dejando la misma distancia desde cada ojo con el borde.
- Se calcula la distancia, coordenada y', para posicionar el rostro correctamente en el eje vertical de la nueva imagen.
- Una vez realizados estos cálculos de alineamiento, redimensionamos la imagen a 200*200.
- Aplicamos un filtrado bilateral, tiene una gran importancia en el procesamiento digital de imágenes ya que posee la ventaja de suavizar las zonas homogéneas de una imagen manteniendo los bordes [19]. Ayudando a reducir la dimensionalidad y eliminar mayor complejidad de calculo.
- La nueva imagen es guardada en un nuevo directorio dentro del actual llamado "modificadas".

Este proceso se realiza con el fin de eliminar elementos innecesario de la imagen, como el fondo y partes del rostro no utilizadas en el proceso de clasificación como el pelo, las orejas, y el cuello. Al eliminar estos elementos innecesarios el entrenamiento se focaliza solo en los componentes fundamentales.

El resultado de este proceso de preprocesado de las imagenes de cada tramo de edad se ve en la figura 4, donde vemos una imagen de un adulto.



Figura 4. Imagen original y la imagen preprocesada

Cuando ha realizado este cometido con todas las imágenes de los cuatro directorios, el siguiente paso es la generación del CSV que utilizaremos para el entrenamiento. El CSV indica al siguiente modulo del sistema donde encontrar las imágenes de cada tramo de edad ya clasificadas y preprocesadas. El CSV indica el directorio donde se encuentra las imágenes de cada tramo, y la etiqueta del tramo de edad.

3.2. AgeTrainer

El segundo subsistema lleva a cabo la tarea de entrenamiento. Con ese fin crea un modelo de reconocimiento FisherFaces, que vamos a entrenar con las imágenes que se encuentran en el CSV creado en el modulo anterior, las cuales llevan asociada la etiqueta de su tramo de edad. Una vez entrenado, se guarda el modelo Fisherfaces en un archivo YML.

Este es el modelo con el que posteriormente se va a comparar para tomar la decisión de cual es el tramo de edad de la persona que estamos viendo por la webcam en tiempo real.

En esta etapa el algoritmo Fisherfaces calcula el rostro promedio que es la media aritmética de todas las imágenes de entrenamiento, por lo que se puede restar esta imagen promedio de cada imagen facial para tener mejores resultados de reconocimiento facial. Esta imagen esta formada tanto por hombres, mujeres y niños 5.



Figura 5. Rostro promedio calculado por Fisherfaces

A diferencia del algoritmo Eigenfaces que busca encontrar componentes para reconstruir la imagen a partir de ellos, y por esto no es indicado para esta tarea, Fisherfaces identifica las regiones de un rostro que separan rostros mejor de los otros. No codifica ajustes de luz particulares, al menos no es tan evidente como en el método Eigenfaces.

3.3. WebCamAgeEstimator

El tercer subsistema, éste el sistema principal que asume la acción de reconocimiento, en la figura 6.

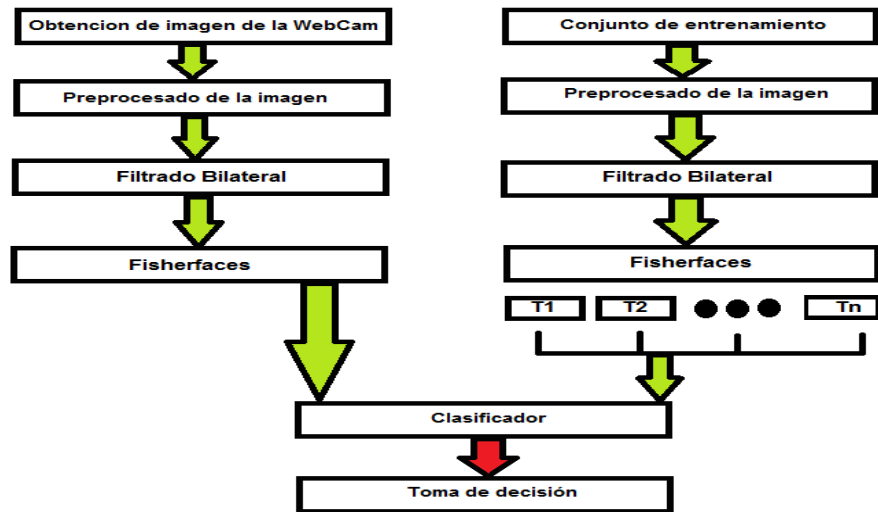


Figura 6. Estructura de funcionamiento de WebCamAgeEstimator

Se inicia con la carga el modelo de entrenamiento de Fisherface, con el que se comparara cada imagen obtenida mediante el sistema de cámaras. Se carga el clasificador en cascada con el cual detectamos los ojos de la persona. Aquí se realiza de nuevo el mismo proceso que en la preparación de la base de datos para el entrenamiento. Detectados los ojos se comprueba si es necesario invertir la imagen, se realiza un centrado de la imagen basado en el punto medio de los ojos, y se redimensiona a 200*200 de forma que al aplicar el algoritmo Fisherfaces clasifiquemos la imagen según se ajuste al modelo de los distintos tramos del entrenamiento.

4. Resultados

En el caso de estudio propuesto se han establecido los siguientes tramos de edad:

- Niños (0 - 18)
- Jóvenes (19 - 30)
- Adultos (31 - 70)
- Ancianos (70 - ND)

Los rostros que conforman estos tramos de entrenamiento y clasificación, son todos rostros en perfil frontal. Como la base de datos de FERET no contiene imágenes de niños estas han sido recogidas manualmente, por lo que la cantidad de rostros no es demasiado elevada.

El sistema realiza la estimación de la edad en un intervalo de diez veces por cada sujeto que aparece en la imagen de entrada, de esta forma el tramo que obtenga la mayoría en las diez estimaciones será el tramo escogido haciendo que el sistema tenga un mayor grado de éxito en tiempo real. Para comprobar la tasa de acierto del sistema de estimación con los datos con los que se ha realizado el entrenamiento se ha realizado una validación cruzada.

Esta prueba se ha realizado tanto con Fisherfaces como con Eigenface y Local Binary Pattern (LBP), y aplicando y no un filtrado bilateral tanto en los datos de entrenamiento como de testeo. Se ha podido comprobar como el método Eigenfaces no es aceptable en tareas de clasificación relacionadas con la edad, debido a que tiene en cuenta factores de ajustes de luz. Mostrando la tasa de acierto en la tabla 1, y certificando el método escogido para la arquitectura del sistema.

	Sin filtro	Filtro Bilateral
Fisherfaces	62,85 %	71,4 %
Eigenfaces	57,14 %	65,71 %
LBP	54,2 %	57,14 %

Cuadro 1. Tasa de acierto

La tabla 1 muestra Fisherface como el método más adecuado, y como al aplicar un filtrado bilateral el índice de acierto incrementa ligeramente. A continuación se muestran las matrices de confusión de los distintos métodos.

Como se puede apreciar en las distintas matrices de confusión, donde se produce menor índice de acierto es en la estimación de jóvenes y de adultos. Siendo bastante llamativo que la estimación del tramo de edad de los niños tenga un gran índice de acierto en cualquier método teniendo la base de datos un número muy inferior respecto de los otros tramos de edad.

	Niños	Jóvenes	Adultos	Ancianos
Niños	3	0	2	0
Jóvenes	0	10	0	0
Adultos	0	2	8	0
Ancianos	0	2	4	4

Cuadro 2. Matriz de confusión de Fisherfaces

	Niños	Jóvenes	Adultos	Ancianos
Niños	4	0	0	1
Jóvenes	0	8	2	0
Adultos	0	3	6	1
Ancianos	0	3	2	5

Cuadro 3. Matriz de confusión de Eigenfaces

	Niños	Jóvenes	Adultos	Ancianos
Niños	4	0	0	1
Jóvenes	0	5	5	0
Adultos	0	0	7	3
Ancianos	0	1	6	3

Cuadro 4. Matriz de confusión de LBP

5. Conclusiones

En la presente memoria se ha demostrado como es posible realizar una clasificación a partir de imágenes faciales de personas tomadas por video cámara en tiempo real. Para ello solo ha sido necesario crear un sistema que realiza la clasificación basado en la extracción de características y el cálculo de los eigenvalues respecto de los tramos de edad propuestos ya entrenados previamente.

A pesar de los esfuerzos de la comunidad científica por desarrollar nuevos algoritmos basados en nuevas formas de extracción de características, cálculos biometricos y complementación entre diversos tipos de sistemas y algoritmos, resulta evidente que Fisherfaces con una base de datos amplia y bien organizada ofrece resultados aceptables, los cuales incrementan ligeramente con el uso de filtros.

La utilización de distintos filtros ha permitido confrontar los resultados y entender que tipo de filtrado ayudaba en la consecución del rendimiento deseable. El filtrado Gaussiano y el filtro Bilateral provocan que al aplicar una técnica de reducción de la dimensionalidad se obtengan menos dimensiones, así Fisherfaces reduce la carga de trabajo, hecho que se agradece en un sistema que realiza la clasificación en tiempo real y constantemente tiene imágenes que computar y no deben acumularse.

Aunque en el entrenamiento se cuenta con imágenes de niños, no en la cantidad necesaria para un buen entrenamiento, no se ha probado este tramo de

edad debido a los problemas de privacidad y protección infantil, siendo necesario permiso paternal.

Uno de los problemas que se han encontrado en la investigación es la clasificación de personas con edad comprendida en el límite de los tramos propuestos, siendo clasificada en ambos grupos. Una posible solución ha sido preparar la base de datos de rostros eliminando caras con edades que pudieran estar en ambos grupos, aunque no es tarea sencilla ya que como se ha aclarado anteriormente, la clasificación de las imágenes de la base de datos se ha realizado con criterio personal basado en la experiencia.

5.1. Líneas futuras de investigación

El trabajo realizado abre muchas líneas nuevas, desde mejorarlo en si mismo hasta desarrollar un nuevo sistema apoyado en el actual. A continuación se citan algunas posibles líneas futuras de investigación.

Complementar Fisherfaces con SVM entrenadas en la extracción de características faciales. La aplicación de un filtro de Gabor para extraer las arrugas faciales y la dirección y un filtro de Sobel para obtener el grado de incidencia de las arrugas. Estudiar y probar distintos métodos de clasificación: K-nearest neighborhood, árboles de decisión, redes bayesianas y/o redes neuronales.

Mejoras relacionadas con la base de datos, como una mejor preparación de la base de datos, con un mayor equilibrio entre etnias. Y una mejor preparación del tramo de edad infantil, con una mayor cantidad de imágenes para el entrenamiento.

Bibliografía

- [1] A. Lanitis, “Facial age estimation,” 2010.
- [2] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung, “A ranking approach for human age estimation based on face images,” 2010.
- [3] G. G. Y. Fu and T. Huang, “Age synthesis and estimation via faces: a survey,” *Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 1955–1976, 2010.
- [4] Y. H. Kwon and N. da Vitoria Lobo, “Age classification from facial images,” in *In Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 762–767, 1999.
- [5] A. Lanitis and C. T. y T. Cootes, “Toward automatic simulation of aging effects on face images,” *Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 442–455, 2002.
- [6] G. Guo, G. Mu, Y. Fu, and T. Huang, “Human age estimation using bio-inspired features,” 2009.
- [7] C. Shan, “Learning local features for age estimation on real-life faces. in mpva’10,” 2010.
- [8] D. Delgado-Gomez, J. Fagertun, B. Ersbøll, F. M. Sukno, and A. F. Frangi, “Similarity-based fisherfaces,” *Pattern Recognition Letters*, vol. 30, no. 12, pp. 1110–1116, 2009.
- [9] M. M. Dehshibi and A. Bastanfard, “A new algorithm for age recognition from facial images,” 2008.
- [10] N. Seo, “Eigenfaces and fisherfaces,” 2007.
- [11] S. E. Choi, Y. J. Lee, S. J. Lee, K. R. Park, and J. Kim, “Age estimation using a hierarchical classifier based on global and local facial features,” 2010.
- [12] R. Jana, H. Pal, and A. R. Chowdhury, “Age group estimation using face angle,” 2012.
- [13] N. Ramanathan and R. Chellappa, “Modeling age progression in young faces,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 387–394, June 2006.
- [14] D. Swets and J. Weng, “Using discriminant eigenfeatures for image retrieval, iee trans.,” 1996.
- [15] P. Belhumeur, J. Hespanha, and D. Kriegman, “Eigenfaces vs. fisherfaces: recognition using class specific linear projection, iee trans.,” 1997.
- [16] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” 1998.
- [17] K. Ricanek and T. Tesafaye, “Morph: a longitudinal image database of normal adult age-progression,” in *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pp. 341–345, April 2006.
- [18] A. Lanitis, “Comparative evaluation of automatic age-progression methodologies,” *EURASIP Journal on Advances in Signal Processing*, vol. 2008, 2008.
- [19] G. Bellino, F. Fernandez, and G. Scarel, “Filtrado bilateral,” 2007.

Sistema para la Distinción de Genero en Imágenes

Francisco Javier Esteban Vicente¹ y Juan Manuel Corchado Rodríguez¹

¹Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España
{javiev, corchado}@usal.es

Resumen La creación de sistemas de reconocimiento de patrones es quizás uno de los mayores retos a los que se ha enfrentado la informática en los últimos tiempos. En este proyecto se pretende estudiar la aplicación del reconocimiento de patrones en cuanto a la distinción del género en imágenes. El alcance de este proyecto estará limitado al estudio en general del estado del arte de esta área y, en particular, al estudio más exhaustivo de dos técnicas: las Fisherfaces y las Redes Neuronales Artificiales. Ambas técnicas tienen en común que se basan en imágenes homogeneizadas mediante la alineación de los rostros para conseguir abstraer las características que les permitan clasificar correctamente. Los objetivos principales serán el estudio y desarrollo de un sistema capaz de distinguir el género en imágenes y tratar de estimar cuanto influye la etapa de preprocesado en estas técnicas.

Keywords: Reconocimiento de Género, Patrones, Imágenes, Redes Neuronales, Fisherfaces

1. Introducción

Desde la década de los 90, uno de los mayores campos de investigación en cuanto al reconocimiento de patrones ha sido la clasificación en géneros a partir de imágenes del rostro. En ocasiones también se han utilizado otras características como la ropa o el pelo [1]. Se han realizado muchas aproximaciones distintas a la hora de tratar de resolver este problema: desde usar redes neuronales pasando por la identificación de características e incluso diversas técnicas matemáticas que permiten reducir la dimensión de las fotografías facilitando así los cálculos. A pesar de que los avances han sido numerosos en todas estas citadas tecnologías, aún es posible mejorar estas técnicas que, en general, requieren de una serie de condiciones controladas para lograr una efectividad alta. La complejidad de este tipo de problemas hace posible el trabajo incluso dentro de campos ya investigados. El número de variables que pueden influir en el proceso hace posible la mejora en casi cualquier tecnología ya probada.

En este trabajo se pretenden estudiar dos técnicas de reconocimiento de género en imágenes con un alto grado de desarrollo: las Fisherfaces[1] y las Redes Neuronales[2, 3, 4]. Las Fisherfaces son una derivación de uno de los métodos de análisis de componentes principales más conocidos, los EigenVectors[1]. Sin

embargo, esta técnica está especializada de tal manera que permita una mejor clasificación entre clases usando de forma efectiva las características que diferencian las clases entre sí, a la vez que minimiza la influencia de las propiedades que diferencian individuos de una misma clase. Las redes neuronales, al contrario que el otro método elegido, si bien dependen completamente de una gran cantidad de cálculos y algoritmos matemáticos no tienen su origen, al menos como idea, en el campo de las matemáticas sino en el de la biología. Durante este trabajo se realizará un estudio comparativo de ambos métodos bajo diferentes situaciones para analizar la eficiencia de cada uno de ellos. Para realizar la comparativa se aplicará sobre un caso de estudio basado en un subconjunto de la base de datos FERET. La comparativa será entre los dos métodos, así como entre 4 sets distintos de imágenes preprocesadas de distinta forma, en concreto cambiando la forma de recortar el óvalo facial. En la primera parte de esta memoria se explicará el estado del arte, seguido de la propuesta realizada por el autor. Los resultados y conclusiones comprenderán el siguiente apartado, para finalizar con las posibles líneas futuras de investigación.

2. Estado del Arte

El reconocimiento facial ha tenido un auge en los últimos años. Distintas técnicas han sido desarrolladas basadas en muy diversas formas de manipular las imágenes para obtener un resultado lo más robusto posible [5, 1, 6, 7, 3, 4, 2, 8, 9, 10, 11, 12, 13]. A pesar de la gran variedad de métodos y avances en este campo, la gran mayoría de ellos tienen algo en común: dependen de un proceso de aprendizaje o entrenamiento, una etapa crucial que será la que defina el posterior rendimiento de nuestro clasificador. Podemos ver en la figura 1 la definición del sistema básico usado para clasificar género en imágenes [5].

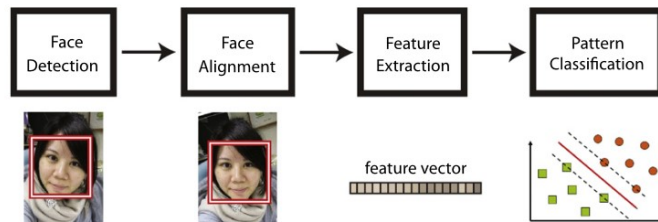


Figura 1. Pipeline para la clasificación de género en imágenes [5]

Unas de las primeras aproximaciones usadas para el reconocimiento de rostros son los métodos de correlación o comparación de plantillas. La forma básica llevaría a cabo una comparación con una base de datos de entrenamiento para clasificar la imagen con el vecino más cercano en el propio espacio de la imagen [6]. Distintas variantes de esta estrategia podrían ser el uso de diferentes plantillas

de cada uno de los sujetos, bien cambiando la posición o tomando solo partes significativas del rostro. Esta técnica puede tener problemas como los propios autores, Brunelli y Poggio, indican en el momento en que las imágenes no estén tomadas con el mismo foco de luz principal[6], cosa que no ocurre con otros métodos como las Fisherfaces [1]. Otros métodos usarán la integral de Fourier para hacer la comparación en el dominio de la frecuencia, haciendo de esta manera el reconocimiento independiente de la luz[14]. Si bien estos autores no lo usaron para la distinción entre géneros sino para el reconocimiento facial, posiblemente una adaptación podría funcionar.

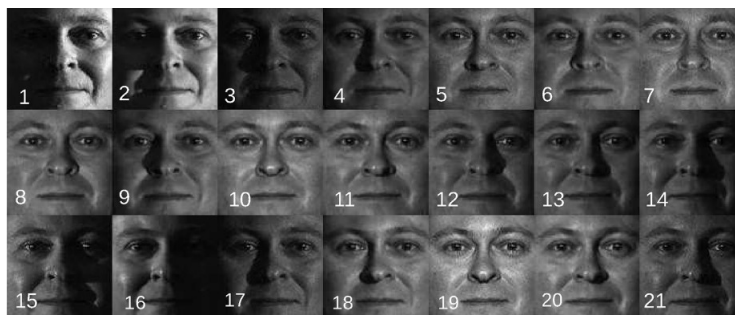


Figura 2. Diferencias provocadas por la iluminación

Debido a que los métodos de correlación o clasificación mediante plantillas tienen una alta complejidad computacional, se ha abordado el problema de la clasificación mediante técnicas de reducción de dimensionalidad, siendo una de las más conocidas la extracción de las Eigenfaces: Reduciendo la dimensión y expresando los datos en función de una nueva base adecuada podemos expresar nuestros datos sin perder información valiosa para la clasificación. La técnica usada para calcular las Eigenfaces, PCA [1], consiste en calcular los elementos principales de la imagen, los llamados Eigenvectors. Los Eigenvectors serán ortogonales entre sí por lo que nos permiten expresar el contenido de una imagen tratada como una matriz en función de sus Eigenvectors usados como base en vez de usar los ejes x e y . Estos Eigenvector serán de dimensión n , con lo que se suele referirse a ellos como Eigenfaces o eigenpictures. El problema de este método es que la variabilidad usada no es sólo la que diferenciará a los elementos de las distintas clases, sino que también se tendrá en cuenta los elementos diferenciadores entre elementos de una misma clase[1], por ejemplo, la iluminación.

Este efecto producido por la luz al usar las Eigenfaces se elimina con el método de las Fisherfaces. La principal diferencia entre ambos métodos es que las Fisherfaces van a aprovechar la base de datos usando no sólo información contenida en las propias fotos sino también los metadatos. El uso de las etiquetas que nos informan de a qué clase pertenece cada imagen ayuda a crear un método más fiable para reducir la dimensionalidad del espacio de características. El método

de las Eigenfaces es una combinación entre las Fisherfaces y el Discriminante Linear de Fisher[15]. El FLD usa una combinación lineal de las características medidas en las clases para clasificarlas. Usando este método conseguimos abstraer nuestras imágenes de la influencia de la luz, en concreto el problema de la dirección del foco principal.

Las redes neuronales también han sido propuestas por diferentes autores a lo largo de los últimos años para el reconocimiento de rostros en imágenes o la clasificación según el género. Mientras que el ejemplo más simple lo tendremos usando una sola red multicapa a la que pasaremos las imágenes, otros más complejos han sido probados con éxito por distintos investigadores. Tenemos por ejemplo, la solución ideada por B. A. Golomb, D. T. Lawrence y T. J. Sejnowski en [3], consistente en dos redes neuronales de 3 capas. En contraste otras aplicaciones anteriores que entrenan la red neuronal directamente a partir de las imágenes de las caras, tenemos la solución propuesta por Brunelli y Poggio[2] que crean un vector de características que será pasado a dos redes del tipo HyperBF que de forma competitiva clasificarán según el género.

3. Propuesta

El objetivo principal de este trabajo es el estudio de distintas técnicas que permitan realizar un reconocimiento de género en imágenes de la forma más robusta posible. Especialmente se estudiarán 2 sistemas. El primer sistema será el uso de las ya mencionadas Fisherfaces. Para su implementación se usará el framework OpenCV. El proceso llevado a cabo para crear un clasificador usando la implementación de Fisherfaces de OpenCV se puede apreciar en la Ilustración 3.



Figura 3. Proceso de Creación de un clasificador usando Fisherfaces

El segundo sistema será una Red Neuronal Artificial. Para la solución de este problema se ha optado por una de las estructuras más conocidas y estudiadas, un perceptron multicapa. La estructura usada es la representada en la Ilustración 4, donde $n = m$ es el número de píxeles de la imagen (4096) y $c = 2$, correspondiente a los grupos a clasificar, hombre y mujer.

En el preprocesado de la imagen se tratan las imágenes en el formato .ppm como una matriz de 64×64 con valores en $[0, 255]$ correspondientes a los valores en escala de grises de cada uno de los píxeles de la imagen. Para prepararlos para la entrada a la red tendremos que normalizarlos en el intervalo $[-1, 1]$.

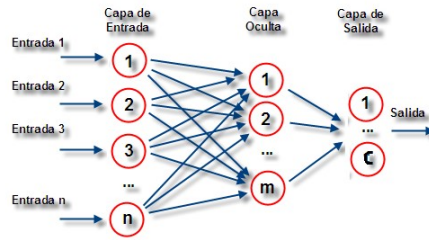


Figura 4. Estructura de la Red Usada

3.1. Preprocesado

El preprocesado es una de las partes más importantes a la hora de hacer funcionar correctamente los clasificadores que se van a usar. A grandes rasgos la etapa de preprocesado de las imágenes comprenderá las fases presentes en la Ilustración 5.



Figura 5. Etapas de preprocesado

A las etapas vistas en Ilustración 5 tendremos que añadir una más para el caso de las redes neuronales, el paso del bitmap de la imagen al vector unidimensional que recibiría la red.

Para la detección de los ojos se usará la implementación de OpenCV de un clasificador en cascada basado en características. Según [16] hay una gran cantidad de razones para utilizar características en lugar de píxeles, entre ellas destaca el hecho de que las características probablemente codificaran conocimiento del set de entrenamiento. Sin embargo, no es este el único motivo sino la velocidad, trabajar con características es muchísimo más rápido que trabajar con sistemas basados en píxeles. Esto se basa en reducir la complejidad computacional representando las características rectangulares mediante su integral. La implementación que OpenCV nos ofrece de estos clasificadores ha sido entrenada con una serie de vistas de un objeto particular (ojos en este caso) escaladas a un mismo tamaño (20x20) junto con una serie de imágenes arbitrarias que permitirán al sistema adaptarse y no dar falsos positivos. Una vez tengamos las coordenadas de los ojos de la imagen, podremos identificar el óvalo facial, recortarlo y girarlo (en caso de ser necesario) para que quede perfectamente alineado a partir de los ojos. El área a recortar será cuadrada, centrada en un punto (X_c, Y_c) de la recta

perpendicular al segmento que une los dos ojos y con origen en el punto medio (Pm_x, Pm_y) de dicho segmento. Además este área estará girada de acuerdo con el ángulo de inclinación de la cabeza (α). Teniendo en cuenta que el sistema de coordenadas es el cartesiano con el punto de origen situado en la esquina superior izquierda, podemos ver como se calcula el centro del área a recortar en la Ilustración 6.

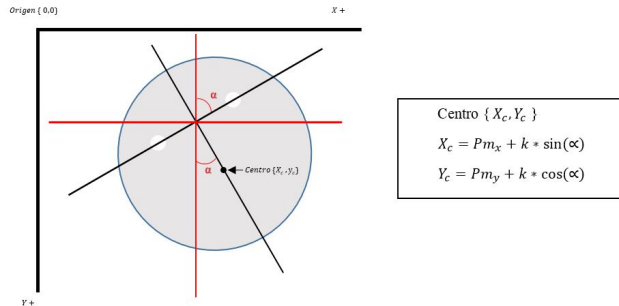


Figura 6. Cálculo del centro del cuadrado a recortar

En la Ilustración 7 podemos observar como pasamos de tener imágenes con diferentes enfoques y tamaños a óvalos faciales completamente homogéneos y normalizados.

Además se aprecia como algunas imágenes necesitarán ser giradas mientras que otras no. El rango de posiciones es amplio sin embargo siempre podremos alinear la imagen con la única condición de que esta esté tomada de frente. De esta forma, podemos conseguir un alineado perfecto con respecto a los ojos, algo indispensable para nuestros métodos de entrenamiento.

Como objetivo de este proyecto se ha postulado desde el principio averiguar la influencia de las distintas formas de hacer el recorte del óvalo facial. En las Ilustraciones 8 y 9 podemos ver el proceso entero de cómo se trata la imagen para obtener finalmente los distintos tipos de recorte variando centro y longitud del lado del cuadrado.

El primer paso, como se ha mencionado anteriormente, es el cálculo del punto medio del óvalo facial. Este paso se realiza a partir de las coordenadas obtenidas al buscar los ojos en la fotografía. Podemos ver en la Ilustración 8 el proceso de giro de la imagen según el ángulo obtenido también a partir de las coordenadas de los ojos.

Finalmente podemos ver en la 9 la forma de recortar las distintas áreas dependiendo del centro y el lado del cuadrado a recortar.



Figura 7. Imágenes antes y después del preprocesado

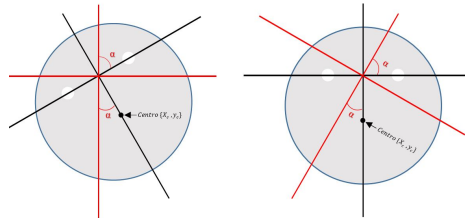


Figura 8. Proceso de giro del óvalo facial

En la Ilustración 10 podemos ver en vertical los 4 recortes realizados para 4 personas diferentes. Se puede apreciar que la diferencia en los rasgos faciales, en concreto el ratio entre la anchura y la altura del óvalo facial, hace que cada sujeto quede mejor enmarcado en uno de los sets de imágenes.

Además, se puede apreciar perfectamente que si cogemos un área mayor tendremos todos los rasgos cubiertos mientras que al coger un área menor, pese a sacrificar parte del óvalo facial, no tendremos trozos del fondo que puedan distorsionar nuestros resultados.

Es por tanto de vital importancia conseguir un equilibrio entre estos dos factores. Este equilibrio será difícil de conseguir teniendo en cuenta, como ya se ha explicado, que hay diferencias apreciables entre sujetos en cuanto al tamaño de la cara, incluso recortando de forma proporcional a los ojos.

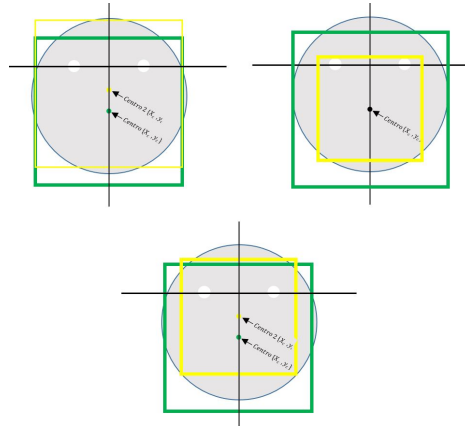


Figura 9. Proceso de recorte del óvalo facial

4. Casos de Estudio

Como base de datos de entrenamiento y prueba se usara la base de datos FERET. El programa FERET estuvo vigente desde 1993 a 1997, financiado por el departamento de defensa a través de la Defense Advanced Research Products Agency (DARPA). Su función principal fue desarrollar métodos automáticos de detección de rostros que pudieran ser utilizados en aplicaciones usadas para ayudar a las fuerzas de seguridad en sus tareas. La base de datos pretende ser una herramienta para evaluar de forma estandarizada distintos algoritmos, para ello cuenta con un total de 14051 imágenes con diferentes puntos de vista. Para la realización de este Proyecto se ha utilizado sólo un subconjunto de la base de datos consistente en las imágenes tomadas de frente y a individuos sin gafas. Esto ha dado un total de 930 imágenes en diferentes tramos de edad comprendidos desde la adultez hasta fotos de ancianos. Los grupos en cada una de las iteraciones para los distintos métodos y sets de imágenes no varían.

Para la creación de la red neuronal se realiza en Java usando el framework Encog. Se usa la red neuronal perceptron multicapa vista en la ilustración 4 con una función de activación Sigmoide y un como método de entrenamiento Resilient Propagation. Como hemos visto anteriormente, el uso de la red neuronal añade una etapa adicional a la hora de tratar con imágenes. Ya que que nuestras imágenes son un bitmap triplicado, debido al formato PPM en el que viene la base de datos, para usarlas como entrada de la red neuronal tendremos que transformarlas en un array unidimensional. Además, no pasaremos los valores en escala de grises directamente sino que los convertiremos en el intervalo $[-1, 1]$. Cada neurona de la capa de salida corresponde a uno de los dos géneros, la salida de la neurona que más se aproxime a 1 será la que marque el sexo elegido por la red.



Figura 10. Distintos recortes de un mismo óvalo facial

OpenCV es la biblioteca nos permite crear nuestro clasificador usando Fisherfaces de una forma rápida y sencilla. Mientras que en el caso de las redes neuronales tendremos que transformar nuestra imagen a un vector para poderlo usar como entrada, en el caso de las Fisherfaces OpenCV admite una gran cantidad de formatos de entrada que se pueden usar a la hora de entrenar y usar nuestro clasificador. Para la creación de un programa con el que obtener el resultado del 5x2 Cross Validation sobre los 4 juegos de imágenes tendremos que definir las carpetas e iterar sobre ellas con el método que nos realizará el entrenamiento y testeo.

5. Resultados

El alcance de este estudio se centra en el óvalo facial de los rostros. Si bien seleccionar el óvalo facial de una imagen puede parecer algo sencillo, se pueden apreciar grandes diferencias dependiendo del recorte que se haga. En la fase de recorte de las imágenes se han creado los bancos de imágenes a partir de la base de datos original, en concreto tendremos 4 versiones diferentes de cada una de las imágenes originales resultantes de variar el área a recortar. Durante la fase de testeo de los modelos se ha usado el método 5x2-Cross Validation[17, 18]. Con el uso de este método tendremos la certeza de que los datos obtenidos al probar el modelo no son dependientes de los subconjuntos de entrenamiento y prueba elegidos. Además, nos permite realizar una comparativa en la tercera fase entre los distintos métodos y versiones de recortes del óvalo facial. En este trabajo se ha conseguido realizar con éxito la implementación y prueba de dos de los principales métodos usados actualmente a la hora de abordar la tarea de la clasificación por géneros, Fisherfaces y RNA, cuyos resultados podemos ver en las Ilustraciones 11 y 12.

	Set 1		Set 2		Set 3		Set 4	
	a	b	a	b	a	b	a	b
Iteración 1	0.903226	0.88172	0.896774	0.894624	0.883871	0.888172	0.892473	0.896774
Iteración 2	0.901075	0.886021	0.898925	0.87957	0.901075	0.875269	0.883871	0.890323
Iteración 3	0.892473	0.875269	0.896774	0.883871	0.886021	0.87957	0.883871	0.886021
Iteración 4	0.886021	0.883871	0.883871	0.847312	0.87957	0.858065	0.873118	0.898925
Iteración 5	0.907527	0.8817	0.892473	0.883871	0.886021	0.858065	0.898925	0.892473
Media	0.8898903		0.8858065		0.8795699		0.8896774	

Figura 11. Resultados Fisherfaces - Indice de acierto

	Set 1		Set 2		Set 3		Set 4	
	a	b	a	b	a	b	a	b
Iteración 1	0.866666	0.894623	0.855913	0.888172	0.877419	0.894623	0.864516	0.877419
Iteración 2	0.836559	0.870967	0.868817	0.88172	0.862365	0.858064	0.88387	0.832258
Iteración 3	0.864516	0.860215	0.894623	0.862365	0.88387	0.851612	0.84086	0.836559
Iteración 4	0.853763	0.860215	0.845161	0.886021	0.855913	0.860215	0.868817	0.851612
Iteración 5	0.896774	0.817204	0.847311	0.855913	0.84301	0.832258	0.84086	0.888172
Media	0.862150538		0.868602151		0.861935484		0.858494624	

Figura 12. Resultados RNA - Indice de acierto

Para contrastar los dos métodos entre si usaremos el test de Mann-Withney[19]. El valor del estadístico para muestras de tamaño 10×10 y un nivel $\alpha = 0.05$ es 27. Para cada uno de los sets calculamos el estadístico de cada grupo. En todos ellos obtenemos un valor de $U_1 < U_2$, el valor útil es el menor de ellos, por tanto la hipótesis formulada para cada uno de los sets de imágenes es la siguiente:

H_0 - La primera población toma valores menores o iguales que la segunda

H_1 - Los valores de la segunda población son menores

Puesto que $U_1 < U_{n_1; n_2; \alpha}$, se cumple en todos y cada uno de los set de datos podemos concluir que los datos obtenidos en la segunda muestra son siempre menores. Por tanto, el mejor método que podemos usar en todos y cada uno de los set de datos creados para lograr una clasificación lo mejor posible es el primer método estudiado: Fisherfaces.

Para contrastar los distintos sets de imágenes dentro de cada uno de los métodos usaremos el test de Kruskal-Wallis [20] aplicado a los resultados de los sets de imágenes de cada uno de los métodos. Para comenzar establecemos un nivel $\alpha = 0.05$ y planteamos nuestra hipótesis.

H_0 - Las 4 muestras no tienen diferencias significativas

H_0 - Las 4 muestras tienen diferencias significativas

Y calculamos los estadísticos, $H_{Fisherfaces} = -10,591463$ y $H_{RNA} = -4,3119512$. Comparándolos con una tabla podemos ver que el valor obtenido es mucho menor que el tabulado, con lo cual aceptamos la hipótesis nula. Por tanto se puede

afirmar que entre los distintos sets de imágenes creados para la prueba no hay una diferencia significativa ni aplicando el método de Fisherfaces ni usando la red.

5.1. Evaluación Temporal

Una de las observaciones realizadas a la hora de probar ambos procedimientos es la gran diferencia en cuanto a la escala temporal a la hora de realizar los entrenamientos de los dos métodos.

	Fisherfaces – OpenCV	RNA – Encog
Set 1	42 minutos	13 horas 23 minutos
Set 2	46 minutos	15 horas 11 minutos
Set 3	41 minutos	13 horas 55 minutos
Set 4	39 minutos	16 horas 2 minutos

Figura 13. Tiempo de entrenamiento y Test para cada una de los sets de datos según el método usado

Podemos ver en la ilustración 13 una comparativa del tiempo que han tardado ambos métodos en realizar esta fase. Por supuesto ambos entrenamientos se han realizado en un mismo equipo.

6. Conclusiones

Tras finalizar este proyecto de investigación podremos llegar a la conclusión de que la etapa de preprocesado es muy importante a la hora de usar los dos métodos elegidos. Al contrario que en otros métodos desarrollados por distintos autores, estos dos clasificadores son totalmente dependientes de esta etapa. Sin embargo, a la luz de los resultados podemos apreciar que no hay una gran diferencia entre usar un juego de imágenes u otros. En cuanto al uso de un método u otro, las Fisherfaces han demostrado ser sensiblemente más eficaces que las redes neuronales. Además, el tiempo de entrenamiento es mucho menor en Fisherfaces. Los resultados globales obtenidos, si bien no son tan altos como algunas de las implementaciones creadas por algunos autores, ofrecen un buen punto de partida sobre el que trabajar para mejorar nuestros clasificadores. En resumen, si bien está claro que la etapa de preprocesado es vital para conseguir un buen funcionamiento de los dos métodos propuestos, los resultados muestran que no son tan sensibles a los pequeños cambios a la hora de realizar esta etapa como se podría pensar en un principio. En lo referente a los dos métodos usados, las Fisherfaces han demostrado ser algo mejor que las RNA.

De los resultados obtenidos al comparar los distintos sets de imágenes creados para este proyecto se puede comprobar que todos y cada uno de ellos llegan a un grado de abstracción suficientemente bueno. De aquí podemos concluir que

los rasgos necesarios para crear un clasificador están contenidos en cada uno de estos sets. Por tanto, crear distintas áreas de recorte conteniendo solo alguno de las características común en todos los juegos de imágenes puede ser más que suficiente para crear un nuevo clasificador, descartando así información posiblemente superflua que pueda ser perjudicial. El uso de la base de datos FERET y las diferencias anatómicas desprendidas de la variabilidad racial pueden hacer a veces difícil la tarea del reconocimiento de género incluso para los humanos, no hay que descartar por tanto que esto puede ser influyente a la hora de crear nuestro clasificador. Distintas soluciones pueden ser dadas a esto: desde la ya mencionada reducción del área a recortar quedándonos solo con los atributos más diferenciadores, al uso de distintos clasificadores según la procedencia del sujeto. Otra posible línea que podría ofrecer una mejoría con respecto al método usado es la combinación de distintos métodos de clasificación.

Bibliografía

- [1] J. P. H. P. N. Belhumeur, D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," 1997.
- [2] T. P. R. Brunelli, "Hyberbf networks for gender classification."
- [3] T. J. S. B. A. Golomb, D. T. Lawrence, "Sexnet: A neural network identifies sex from human faces," 1991.
- [4] R. R. E. Makinen, "An experimental comparison of gender classification methods."
- [5] C. S. C. W. S. Chu, C. R. Huang, "Gender classification from unaligned facial images using support subspaces," 2013.
- [6] T. P. R. Brunelli, "Face recognition: Features vs. templates," 1993.
- [7] S. U. Y. Adini, Y. Moses, "Face recognition: The problem of compensating for changes in illumination direction."
- [8] B. L. B. Li, X. Lian, "Gender classification by combining clothing, hair and facial component classifiers," 2012.
- [9] L. B. J. Bekios-Calfa, J. M. Buenaposada, "Robust gender recognition by exploiting facial attributes dependencies," 2013.
- [10] Z. G. S. Y. H. Kim, D. Kim, "Appearance-based gender classification with gaussian processes," 2005.
- [11] E. H. J. Wu, W. A.P. Smith, "Facial gender classification using shape-from-shading."
- [12] L.-B. L. J. Zheng, "A support vector machine classifier with automatic confidence and its application to gender classification," 2010.
- [13] K. K. R. Preeti, "A gender classification system robuts to occlusion using gabor features based (2d) pca," 2014.
- [14] P. K. B. y A. K. Datta, "Class specific subspace dependent nonlinear correlation filtering for illumination tolerant face recognition."
- [15] R. A. Fisher, "The use of multiple measurements in taxonomic problems," 1936.
- [16] R. J. Baron, "Mechanisms of human facial recognition."
- [17] A. F. J. Camacho, "Cross-validation in pca models with the element-wise k-fold algorithm: Practical aspects."
- [18] "Cross validation." <http://www.cs.cmu.edu/~schneide/tut5/node42.html>. Accessed: 07/2014.
- [19] "Contraste de mann-withney." <http://www.bioestadistica.uma.es/libro/node154.htm>. Accessed: 07/2014.
- [20] "Contraste de kruskal-wallis." <http://www.bioestadistica.uma.es/libro/node156.htm>. Accessed: 07/2014.

Arreglando y analizando textos

Antonio Juan Sánchez¹, Emilio S. Corchado Rodríguez¹ y Sara Rodríguez González¹

¹Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España
{anto,escorchado,srg}@usal.es

Resumen En este artículo se presenta un método para la reconstrucción de los textos con formato incorrecto en español. Está orientado sobre todo a textos que pueden encontrarse normalmente en redes sociales. También se presenta un método de análisis de los sentimientos basado en el análisis de pares. Este método de análisis de sentimiento incluye potenciadores y aplica los mismos procedimientos para analizar las palabras, frases y párrafos. El método se centra en español pero el tipo de solución se puede extender a cualquier idioma..

Keywords: Análisis de textos, Reconocimiento del lenguaje, Análisis de los sentimientos, Minería de opiniones.

1. Introduction

En entornos como redes sociales, foros o bloques de comentarios de páginas web el uso del léxico y la gramática no es del todo correcto, o más bien se puede decir que no es estándar. Las deformaciones de palabras y abreviaturas que no se incluyen en el uso normal de la lengua se producen continuamente. Estas deformaciones pueden incluso mutar de una manera muy específica en función de la zona del país, de acuerdo con las diversas peculiaridades de las variaciones regionales de la lengua.

Este fenómeno no está cerrado a un solo idioma, como se espera, se produce regularmente en todos los principales idiomas utilizados en los entornos sociales.

El estudio de Almansa, Fonseca y Castillo [1], se centra en la variabilidad en el uso de lenguaje de España y América del Sur, y el estudio de Alba Torrego [2] sobre las variaciones léxicas en el entorno Tuenti describe cómo esas variaciones cambian en función de la edad (según Arroyo [3]) y la localización (también en [1]). El proceso de la evolución del lenguaje es un ciclo puramente biológico, guiado por la selección natural, donde las posibilidades de éxito y aceptación determinan la supervivencia de los nuevos términos dentro de la comunidad. Esto hace que este modelo sea difícil de predecir, y la única alternativa parece la adaptación continua.

2. Estudio de la variabilidad del lenguaje

Como ya se ha comentando, el estudio [2] nos da las pistas sobre que variaciones sobre el lenguaje estándar que podemos encontrarnos en las redes sociales. Los casos más generales son los siguientes (la mayoría de ellos son recogidos del estudio de [4]):

- Calcos lingüísticos no reconocidos por la RAE, como "family", "party", "kiss", "tuto", "belo", "muito". Lo más frecuente es que estos calcos vengan del inglés o de lenguas latinas como italiano, portugués o francés, y sean de uso frecuente en la lengua hablada, pero no reconocidos por la RAE.
- Onomatopeyas con sentido nominal, adverbial o adjetival, como por ejemplo "buuff" (designa aburrimiento), "grrrr" (designa enfado), "tic tac" (para designar tardanza), "muac" (para enviar un beso).
- Emoticonos. Los emoticonos son definidos por [Yus, 2010] como "combinaciones de signos de puntuación simples cuya unión parece formar diferentes expresiones de la cara y otros signos icónicos". Su significado, al igual que las onomatopeyas, puede ser muy variado, aunque su uso más común es la representación de estados de ánimo.
- Se pueden encontrar siglas de abreviaciones de palabras de uso común en inglés como "WTF" (What the fuck?) o "LOL" (laughing out loud).
- Existe también una relación de expresiones de uso común en castellano, como "bn" (buenas noches), "tq" (te quiero) que se abrevian con siglas. A menudo existe una alta posibilidad de polisemia, puesto que en número de combinaciones de letras en grupos de 2 o 3 resulta frecuentemente limitado, y su significado es atribuido a través del contexto de la oración.
- Acortamiento de palabras, como por ejemplo "peli.^{en} lugar de "película", "xtremo.^{en} lugar de ".extremo". Esto también es muy común en la lengua hablada.
- Sustituciones de sílabas por dígitos, como por ejemplo en "to2" (todos). Estas sustituciones se deben sobre todo a la equivalencia fónica entre la expresión numeral y la sílaba sustituida.
- Deformaciones de palabras, como "tronxo.^{en} lugar de "troncho", "desfaxar.^{en} lugar de desfasar. Estas deformaciones pueden darse tanto en palabras del idioma patrio como en los calcos lingüísticos que se han visto anteriormente.
- Es también resaltable el uso de signos de puntuación como la exclamación y la pregunta a modo de intensificadores ("¡!!!!.º ¿????"). Este uso está muy vinculado a su utilización ahora muy común en cómics y tiras de prensa.
- Existen malformaciones de palabras y palabras recortadas en su extremo anterior, posterior o incluso en el interior de las mismas. Estos recortes son muy habituales dentro del idioma y se derivan de las relajaciones vocálicas de uso hablado de la lengua, como por ejemplo ".and-ao.^{en} lugar de ".andado", "çom-ío.^{en} lugar de "çomido", ".es- t-do.^{en} lugar de ".es-ta-do.º "dstinto.^{en} lugar de "distinto". La mayoría de estas malformaciones ortográficas tienen patrones comunes y han sido ampliamente estudiadas e identificables.

2.1. Propuesta de algoritmo de reconstrucción de textos

De acuerdo los casos vistos anteriormente, se han acotado los problemas de la variabilidad del uso del lenguaje a los siguientes supuestos: Existen palabras, expresiones o grafemas que son totalmente nuevas con respecto al idioma de uso estándar, como es el caso de las onomatopeyas, emoticonos, siglas de expresiones en distintos idiomas, o calcos lingüísticos. Para estos casos se utilizará una sustitución directa por una palabra o una expresión. Es decir se creará un diccionario de términos de sustitución.

- ;) := "Guiño"
- ! := ".^{En}fadado"
- party := "Fiesta"
- tq := "te quiero"
- bn := "Bien/ "Buenas noches"
- WTF =>"Sorprendido"

Este es el caso más habitual. Existe otro caso en que la palabra analizada simplemente es una mutación de alguna palabra ya existente. En este caso se trataría de forma particular la irregularidad a través de la identificación y sustitución de prefijos, sufijos e infijos irregulares. Ejemplos:

- -ao := -dao
- x- := ex-
- n- := en
- -ds- := -dis-
- k- := que
- -q- =>que

Sobre una palabra se pueden aplicar una o varias transformaciones, dependiendo del grado de degradación de ésta:

- kdao := quedado
- smrart := esmerarte

Mediante este método se podría dar solución a todas las "deformidades" comentadas en el con anterioridad, entre ellas, el uso de expresiones numéricas dentro de los textos que sustituyen a sílabas. La sustitución de palabras puede dar problemas debido la polisemia intrínseca de los términos utilizados, como en el ejemplo de "bn", por ello se debe tener en cuenta el contexto a la hora de hacer las sustituciones. En el caso comentado de "bn" solo se realizaría la sustitución si en el contexto de la palabra se encontraran las palabras "noche", "dormir", "nocturno", "çena", etc. Teniendo en cuenta las variaciones lingüísticas vistas se propone un sistema de reconstrucción de texto basado en el uso combinado de soluciones léxicas y semánticas. Aplicando por un lado un diccionario sustitución de palabras y, por otro, un diccionario de sustitución de morfemas (diferenciando si estos van al principio, en el medio o a final de palabra). En el

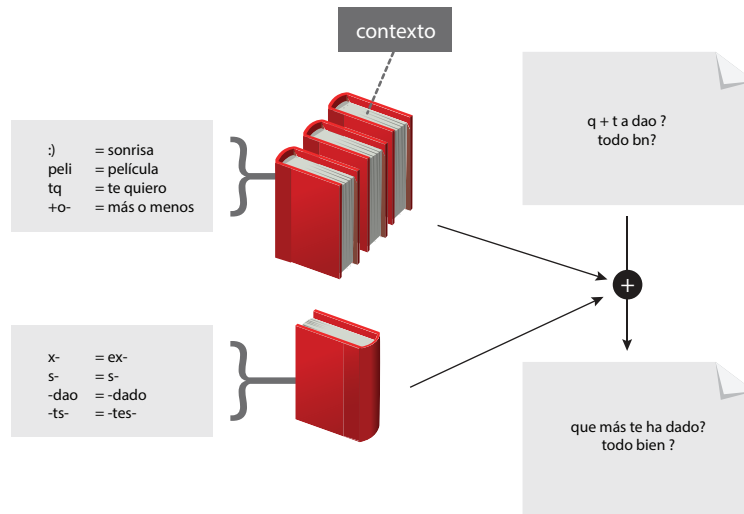


Figura 1. Aplicación de diccionarios

caso de encontrar una sustitución directa, el segundo caso no se aplicaría, puesto que se supone que la sustitución es correcta.

Para solucionar el problema de la polisemia de términos reducidos se propone incluir como criterio para las sustituciones un factor de contexto, es decir, proponer un contexto (una región de texto que rodea a la palabra evaluada) que deba cumplir ciertas propiedades. Esas propiedades determinarán la inclusión de otros términos que se encuadren dentro del campo semántico de la palabra sustituida.

En la 2se puede ver el pseudocódigo para solucionar el problema antes descrito.

```
Dictionary
{
  myDictionary : HashMap
  infixs: HasMap
  getWordsPossibilities (searchedword: String, context:String)
  {
    returnvalue -> {}
    if searchedword is into myDictionary:
      for wordincontext in context
        if wordincontext is myDictionary[searchedword]["context"]:
          returnvalue -> returnvalue
            + myDictionary[searchedword]["value"]
    return returnvalue
  }

  applySustitutions (searchedword: String)
  {
```

```

    returnvalue -> {}
    for (infix, infixvalue) in infixs
        if searchedword contains infix:
            returnvalue -> returnvalue + replaceInfix (searchedword, infix, infixvalue)
    return returnvalue
}

}
searchedword: String;
context:String
posibilities = Dictionary->getWordsPosibilities (searchedword, context)

if (empty (posibilities) == true):
    posibilities -> Dictionary.applySubstitutions ()

```

En el pseudocódigo no se especifica de forma explícita la función `replaceInfix`, que tendría un comportamiento distinto para infijos, sufijos y prefijos. El proceso de reconstrucción del texto depende del tratamiento que se le quiera realizar más tarde, es posible que se necesite un pre-análisis para saber si la reconstrucción resulta rentable computacionalmente hablando o no. Este pre-análisis puede realizarse, por ejemplo a través de un recuento de palabras a reconstruir que no se encuentren dentro del lecionario de español. Lo óptimo sería que además de todo lo anterior se incluyera dentro de un proceso iterativo que generará todas las posibles situaciones de sustitución de palabras y seleccionara un criterio para decidir cuál es la más correcta. La propuesta al respecto para este proceso es la hacer una análisis gramatical de la frase generada y compararlo con el análisis de una frase bien formada extraída de un corpus en español. La proximidad (entiéndase por proximidad la similitud entre las estructuras gramaticales) de la frase a las estructuras del corpus bien formado, dará la calidad de frase obtenida tras hacer las sustituciones.

2.2. Propuesta de algoritmo para análisis de sentimiento basada en pares

La propuesta de este de análisis de sentimiento está basada en un análisis palabra a palabra, incluyendo el uso de intensificadores y atenuadores. El esquema de análisis se amplía desde las palabras hasta los párrafos. Dado que previo al proceso de análisis de sentimientos se ha realizado una "traducción" española estándar, no es necesario tener en cuenta problemas gramaticales o léxicos. Previamente a la aplicación del algoritmo la primera acción es dividir el texto analizado en frases, da igual que éstas sean ordenadas o subordinadas. A continuación se realizará un pre-procesamiento del texto en el que se sustituyan las sentencias adverbiales por adverbios, para poder hacer un análisis palabra a palabra (por ejemplo "mejor que" por "mejor") esto facilitará la detección de la polaridad exacta de las expresiones (entiéndase por polaridad una expresión "positiva" o "negativa"). Además, toda palabra que no tenga un valor relevante

dentro del análisis (como los adverbios, pronombres, artículos o conjunciones) se elimina del texto a analizar para aumentar la velocidad del procesamiento. El cálculo de la polaridad de la frase se llevará a cabo a través de una variante de la fórmula propuesta por Brooke, Tofiloski, y Taboada [Brooke et al., 2009]. La variación sobre este sistema consiste por un lado en la aplicación de las consideraciones de Yann [Yann, 2008] para la modificación de la polaridad, y por el otro lado, la expansión del sistema de análisis de pares al conjunto de frases de un texto hasta llegar al nivel de párrafo. Para hacer el análisis se seguirán las siguientes consideraciones

- Cada bigrama (conjunto de dos palabras) tendrá un valor numérico.
- Una frase de x palabras contendrá $x-1$ bigramas, los bigramas estarán formados por cada palabra de la frase y su palabra anterior.
- En primer lugar se analizaría la polaridad de cada palabra del bigrama que podrá variar entre 1 y -1 atendiendo su valor en una tabla de equivalencias (por ejemplo "bueno" tendría un valor de "0.5", mejor tendría un valor "1-2, malo tendría un valor 0.5.º mejorable tendría un valor 0.2").
- Si una palabra con sentido negativo viene precedida por una palabra de sentido positivo (como por ejemplo "bien malo") ambas cambian su valor a negativo.
- Si una palabra viene dada precedida o sucedida por un intensificador, el valor del intensificador aumentado en una unidad multiplicará el valor de la palabra.
- Los valores de los intensificadores variarán entre -1 y 1. Los intensificadores con valor inferior a 0 serán considerados atenuadores, y aquellos con valor 0 no ofrecerán ningún tipo de intensificación.
- El valor numérico es la suma de los valores de los bigramas que contiene, además cada frase tendrá asociado el valor adicional de la suma de los valores de los intensificadores que la forman.
- Si consideramos una frase como uno de los monogramas que forman un bigrama, con sus valores de polaridad e intensificación el valor numérico que obtiene un párrafo es el resultado de la suma sus bigramas (conjuntos de dos frases).

Formalizando la operación el análisis de un conjunto de bigramas (donde cada palabra tiene valores propios de intensificación y polaridad):

$$t(a, b) = \begin{cases} 1, & a \geq 0, b \geq 0 \\ -1 & \text{i.o.c.} \end{cases}$$

$$u(a, b) = \begin{cases} (a + 1) \cdot (b + 1), & a \neq 1 \text{ or } b \neq 1 \\ 1 & a = 1, b = 1 \end{cases}$$

$$V_F = \sum_{j=2}^N ((M_{j-1}^P + M_j^P) \cdot t(M_{j-1}^P, M_j^P) \cdot u(M_{j-1}^I, M_j^I))$$

$$V_F = \sum_{j=1}^N (M_j^I)$$

Siendo:

- t: Función del cálculo de la intensificación
- u: Función del cálculo del signo de la polaridad
- V_F : Función de cálculo del valor de la polaridad de un conjunto de bigramas
- V_I : Función de calculo del valor de la intensidad de un conjunto de bigramas
- M_j^I : Valor de la intensidad del monograma j
- M_j^P : Valor de la intensidad del monograma j
- N: Número de monogramas

La misma ecuación puede usarse para analizar cualquier estamento, ya sea una lista de palabras, un conjunto de frases o un grupo de párrafos.

Ejemplo de aplicación conjunta de los algoritmos de procesamiento de textos Ambos algoritmos, el de reconstrucción del textos y el de análisis de sentimientos han sido implementados en Python. La información relativa a los valores de las intensidades y de las polaridades han sido codificadas dentro de archivos JSON con la siguiente estructura:

```
{
  "muy": {"multiplicador": 0.5}
  "buenas": {"multiplicador": 0.5}
  "bien": {"sentificador": 0.5},
  "aglegra": {"sentificador": 1},
  "feliz": {"sentificador": 1},
  "muchísimo": {"multiplicador": 1}
  "poco": {"multiplicador": -0.2}
}
```

Donde se indica la palabra y sus valores de intensificación ("multiplicador") y polaridad ("sentificador"), Para la implementación de diccionarios de sustitución de palabras y morfemas se han generado igualmente archivos JSON en los que se indican las sustituciones y el campo semántico del contexto en el que éstas se deben realizar.

```
{
  "bn": [
    {"acepcion": "bueno"},
    {"acepcion": "bien"},
    {"acepcion": "buenas noches",
     "contexto": { "rango": 30,
                   "valores": ["noche", "cama"] } } ],
  "m" : [
    {"acepcion": "me"},
    {"acepcion": "mio"} ],
  "q": [
    {"acepcion": "que"} ]
}
```

Ejemplo de diccionario

Como podemos ver dentro del grupo del contexto "se incluyen dos etiquetas rango", que hace referencia a numero máximo de palabras anteriores o posteriores a la palabra que se desea sustituir, y "valores" que son lo valores que pueden encontrarse dentro del campo semántico. Las clases generadas en Python admiten la inclusión de uno o varios diccionarios de los distintos tipos, e incluyen opciones para comprobaciones estrictas o de contexto. Además se propone identificar el grado en que una frase ha sido reconstruida un análisis léxico de los componentes de ésta, para comprobar si esa misma estructura es usada comúnmente en el idioma español. El funcionamiento completo de todo lo explicado se puede observar en el siguiente ejemplo, que es muy sencillo, donde se han definido como diccionarios para la reconstrucción textual los que están descritos anteriormente, y como texto el siguiente:

"Te e visto muy bn. M aplegra q puedieras venir. ;). M voy a la cama, bn"

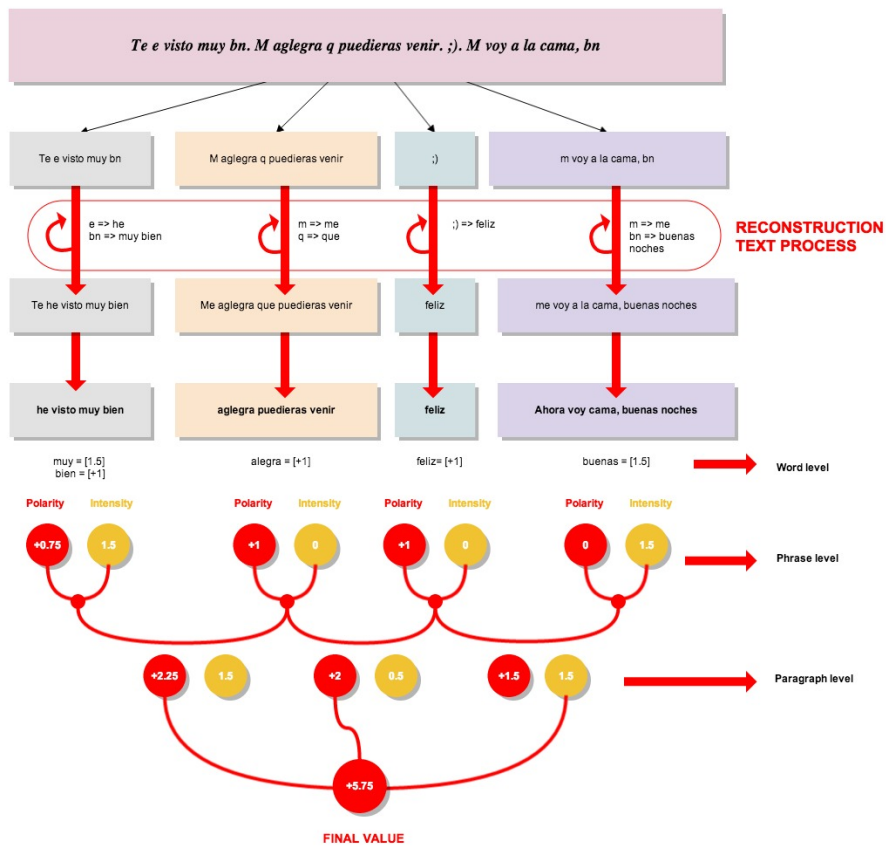


Figura 2. Ejemplo de reconstrucción y análisis de una frase

Agradecimientos Este trabajo ha sido realizado en marco del proyecto PIAR parcialmente subvencionado por MINETUR, TSI-100201-2013-20

Bibliografía

- [1] O. F. A. Almansa and A. Castillo, “Redes sociales y jóvenes. uso de facebook en la juventud colombiana y española.,” *Comunicar*, vol. 40, pp. 127–135, 2013.
- [2] A. Torrego González, “Algunas observaciones acerca del léxico en la red social tuenti,” 2011.
- [3] J. L. B. Arroyo, “La variación léxica,” in *Panorama de la lexicología* (E. de Miguel Aparicio, ed.), Ariel, 2009.
- [4] D. ESCANDELL MONTIEL, “Ciberpragmática en ele. aspectos fundamentales para una comunicación . fiape. iv congreso internacional: La enseñanza del español en un mundo intercultural,” 2011.

Automatic image analyzer to assess retinal vessel caliber (Altair)

Pablo Chamoso Santos¹ and Angélica González Arrieta¹ Sara Rodríguez González¹

¹Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España
{chamoso,angelica,srg}@usal.es

Resumen El examen del fondo de ojo permite realizar una evaluación no invasiva de la microcirculación retiniana, así como de los daños vasculares inducidos por múltiples factores de riesgo cardiovascular. El objetivo de este trabajo es el estudio de las necesidades existentes para dar lugar al desarrollo y validación (fiabilidad y validez) de una metodología que defina un conjunto de técnicas que sean capaces de extraer toda la información presente en imágenes de fondos de ojo para dar solución a las necesidades estudiadas. Su implementación servirá para, posteriormente analizar su utilidad en diferentes entornos clínicos. En la actualidad existen diferentes trabajos propuestos para evaluar el grosor de las venas y arterias retinianas, pero, o bien precisan de intervención total por parte de un observador, o bien no precisan de ninguna intervención, por lo que ante análisis incorrectos (ninguna consigue un 100% de acierto en análisis automáticos) proporcionan resultados erróneos que pueden suponer un grave problema a la hora de extraer conclusiones. La solución planteada define una metodología que compone una serie de técnicas y que entra dentro del grupo de las segundas (automática) pero ofreciendo a un supervisor la posibilidad de interferir en el análisis para los casos en que se produzca cualquier tipo de error, que idealmente serán las menos veces. Con esto se logra que la posible subjetividad que pueda introducir el supervisor, no influya en el resultado del análisis.

Keywords: Análisis de imagen, Fondo de ojo, Retina, Calibre vascular, CBR

1. Introducción

El examen del fondo de ojo permite realizar una evaluación no invasiva de la microcirculación retiniana, así como de los daños vasculares inducidos por múltiples factores de riesgo cardiovascular. Este examen se realiza mediante el análisis de una imagen que se genera con una máquina denominada retinógrafo. Estas imágenes tienen una forma naranja que varía con la piel de color y la edad del paciente. Pero además, esconden en ellas gran cantidad de información que permite detectar enfermedades.

Por ejemplo, en estudios poblacionales realizados, se ha encontrado una asociación del calibre de los vasos de la retina con la hipertensión arterial [1], hipertrofia ventricular izquierda [2], síndrome metabólico [3], accidente cerebrovascular [4] y enfermedad coronaria [5], sobre todo en mujeres [6]. Sin embargo, no todos los estudios coinciden y hay resultados contradictorios en relación a la evolución de la lesión arteriosclerótica y el calibre de los vasos de la retina [7][8][9]. Así, en [10] y en [11], en dos poblaciones de hipertensos, no encontraron asociación entre el calibre de los vasos retinianos y lesiones cardíacas, vasculares o renales, que fueron las analizadas.

Se han desarrollado diferentes metodologías para calcular el grosor de las arterias y venas retinianas, pero presentan algún tipo de carencia. Por este motivo el equipo de investigación médica involucrado en el trabajo, perteneciente a la unidad de investigación de La Alamedilla de Salamanca (Sacyl) creyó conveniente la actualización adaptada a sus necesidades de estas metodologías y trabajos ya existentes.

Por tanto, la metodología propuesta debe ofrecer al equipo de investigación médica la posibilidad de llegar donde otras no ofrecen, de una manera conjunta para satisfacer todas sus necesidades investigadoras. Evidentemente, la metodología planteada debe realizar mediciones de una manera fiable y validada para analizar su utilidad en diferentes entornos clínicos.

Gracias a este amplio conjunto de datos que se espera tener, se puede plantear un sistema experto de razonamiento basado en casos (CBR por sus siglas en inglés Case Based Reasoning) que sea capaz de realizar diagnósticos a partir de una memoria de casos que inicialmente será elaborada por un experto a partir de los datos generados por la herramienta.

1.1. Objetivos

El objetivo general de este trabajo es el desarrollo de una metodología que combine diferentes técnicas algorítmicas y de visión artificial aplicadas a análisis de fondo de ojos. Esta metodología se deberá implementar para lograr su validación (fiabilidad y validez) y extraer información acerca de los vasos sanguíneos presentes en las imágenes a procesar.

Esta información extraída debe, por una parte, contar con aquellos parámetros que clínicamente han demostrado tener relación con enfermedades, patologías o riesgos, como es el caso del grosor medio de las venas y las arterias. Por otra parte, ha de incluir información adicional que pueda suponer clínicamente una novedad. De esta manera se dispondrán de nuevos datos a la hora de realizar estudios en busca de posibles relaciones con problemas no conocidas.

Se realizará adicionalmente un estudio de sistemas CBR (*Case Based Reasoning* o, en español, razonamiento basado en casos), que servirán de ayuda a la hora de interpretar los parámetros generados y buscar asociaciones entre estos parámetros y problemas o enfermedades de los pacientes.

2. Estado del arte

2.1. Visión artificial

Introducción La visión artificial o visión por computador es un campo de la inteligencia artificial que incluye métodos para adquirir, procesar, analizar y comprender imágenes y, en general, los datos de alta dimensión del mundo real con el fin de producir información numérica o simbólica [12]. La visión artificial también ha sido definida recientemente como la empresa de la automatización e integración de una amplia gama de procesos y representaciones de la percepción visual [13]. En [14], se define la visión artificial como una herramienta que permite establecer una relación entre el mundo real (tridimensional) y las vistas bidimensionales tomadas de él. En general, el desarrollo de la visión artificial surge para multiplicar la capacidad de la visión humana por vía electrónica de la percepción y comprensión de imágenes [15].

En nuestro caso nos centraremos en la parte de reconocimiento. Esta tarea normalmente se puede resolver de forma robusta y sin esfuerzo por un humano, pero todavía no puede ser solucionada de manera satisfactoria a través de visión artificial en la mayoría de los casos (objetos arbitrarios en situaciones arbitrarias).

La organización de un sistema de visión artificial, ésta depende de la aplicación. Sin embargo, hay funciones típicas que se encuentran en muchos sistemas de visión por ordenador. La organización más completa de un sistema de visión artificial es:

Adquisición de las imágenes: una imagen digital es producida por uno o varios sensores de imagen que, además de varios tipos de cámaras sensibles a la luz. Dependiendo del tipo de sensor, la imagen de datos resultante es bidimensional, un volumen tridimensional o una secuencia de imágenes [16].

Preprocesamiento: antes de aplicar un método de visión artificial, se pueden aplicar diferentes partes específicas de éstos con el fin de extraer información específica.

Extracción de características: se trata de extraer características de la imagen.

Detección o segmentación: en algún momento de la tramitación, se toma una decisión que acerca de qué puntos de la imagen o de sus regiones son relevantes para su posterior procesamiento.

Procesamiento de alto nivel: en este paso, la entrada suele ser un pequeño conjunto de datos, por ejemplo un conjunto de puntos de una región de la imagen que se supone que contiene un objeto específico.

Toma de decisiones: la toma final de decisiones requeridas para la aplicación, suelen ser, por ejemplo: la aprobación o el rechazo o el marcado para posteriores revisiones realizadas por expertos en los ámbitos médico, militar, de seguridad y aplicaciones de reconocimiento.

Filtros y técnicas A lo largo de este subapartado, se detallarán algunos de los filtros y de las técnicas que, en los trabajos relacionados con el caso de estudio, se proponen y posteriormente se verá con qué finalidad se aplican. Para ello, antes se debe tener claro en qué consiste cada uno.

Normalización de contraste local: existen diferentes metodologías para llevar a cabo esta técnica que se pueden dividir en dos grandes clases: la primera implica la descomposición de la imagen en altas y bajas frecuencias y en la combinación de estas dos señales independientes [17].

Corrección de iluminación: es una técnica muy aplicada en la etapa de preprocesamiento de la imagen. Existen múltiples técnicas para la normalización de la iluminación para condiciones en las que la luminosidad varía, como es el caso de las imágenes de retinografía, aunque la mayor parte de trabajos abordan la problemática del reconocimiento facial. Ejemplos de estas técnicas son la aplicación de la transformada discreta del coseno, como se propone en [18] o nuevamente con filtrados homomórficos como se describe en [19]. Para cambios de iluminación pronunciados, se pueden aplicar algoritmos basados en un filtro de Kalman como se describe en [20].

Reducción de ruido: el ruido generalmente aparece, a la hora de adquirir la imagen. Para mejorar el análisis, durante la etapa de preprocesamiento se recomienda tratar de detectar ese ruido y reducirlo en la medida de lo posible para eliminar información incorrecta que pueda inducir a un análisis erróneo de la información. En [21], por ejemplo, para la eliminación de ruido en imágenes de la retina, se aplica un filtro de media a regiones de 3x3 píxeles. En [22] se propone un algoritmo iterativo para la eliminación del ruido a la hora de hacer evaluaciones clínicas de nódulos pulmonares. Los filtros gaussianos que posteriormente se detallarán y los filtrados borrosos, se utilizan también para este fin [23][24].

Análisis de histogramas: en [25] se propone el análisis de histogramas para detectar el disco óptico, pero también se puede aplicar en la etapa de preprocesamiento para eliminar irregularidades. Estos análisis requieren mucha menos coste computacional que otros algoritmos existentes. Resulta muy útil a la hora de determinar si el contraste o la exposición de una imagen son adecuados. Además, del análisis del histograma deriva la posibilidad de obtener un umbral óptimo para cada región de la imagen.

Transformación divisoria: en [26] se proponen diferentes algoritmos de este método. Tiene unas características interesantes para diferentes aplicaciones de segmentación de imágenes. Es simple e intuitivo, puede ser paralelizado y siempre produce una división completa de la imagen. Sin embargo, en análisis de imágenes médicas tiene alguna desventaja como puede ser la sobresegmentación, su sensibilidad al ruido y la pobre detección en estructuras finas [27].

Umbrales adaptables: es una técnica muy aplicada en el análisis de imágenes de la retina, por ejemplo, algunos trabajos propuestos se tienen en [28] o [29]. El grupo de algoritmos que se pueden aplicar tiene como finalidad segmentar las imágenes.

Transformada de Hough: esta técnica se utiliza para la detección de figuras.

Con ella es posible encontrar todo tipo de figuras que se puedan expresar de manera matemática. En el caso de las retinas, las formas buscadas para la localización de los bordes del disco óptico son circunferencias o elipses. Se puede usar como detector de bordes, obteniendo los puntos de la imagen que pertenecen a la frontera de la figura deseada, por lo que previamente hay que aplicar alguna técnica antes de buscar el disco óptico.

Redes neuronales: existe mucha literatura acerca de las redes neuronales. Lo que se pretende es explicar el motivo de su aplicación en sistemas similares al presente caso de estudio. La mayor parte de esta literatura relacionada tiene que ver con sistemas de detección facial, como el propuesto en [30]. Respecto al análisis de imágenes de la retinas, la utilización de redes neuronales consigue la extracción de características de la imagen para las que se ha entrenado la red, como en [31].

Clasificadores: los clasificadores utilizan algoritmos para asignar a un elemento entrante que inicialmente no se encuentra etiquetado, una categoría concreta conocida. Dichos algoritmos, permiten ordenar o disponer por clases los elementos entrantes a partir de cierta información característica de éstos [32]. Una manera de implementar un clasificador es la selección de un conjunto de ejemplos etiquetados y tratar de definir una regla que permita asignar una etiqueta a cualquier otro dato de entrada.

Filtros gaussianos: utilizan como base funciones gaussianas, logrando un suavizado al aplicarlas a la imagen. Esto se consigue a través de la aplicación de algoritmos matemáticos que en esencia, mezclan ligeramente los colores de los píxeles vecinos, provocando que la imagen pierda detalles y de esta manera, la imagen se ve más suave.

2.2. Sistemas de razonamiento basado en casos

El razonamiento basado en casos (CBR por sus siglas en inglés, *Case Based Reasoning*) es un tipo de razonamiento que el ser humano utiliza en su pensamiento, donde se recurre a experiencias pasadas para solucionar nuevos problemas [33]. Esto es así debido a que se tiene que en un momento pasado se decidió resolver un problema aplicando una solución o un conjunto de soluciones basados en ciertos criterios y, al aplicarla, el resultado obtenido fue bueno o, por lo menos, aceptable. Por eso parece lógico que, en caso de volver a aplicar la misma solución cuando se vuelva a plantear un caso con un problema de las mismas (o similares) características, se tenga nuevamente un resultado del mismo estilo.

Por el contrario, si el resultado obtenido en el pasado fue malo o no todo lo bueno que era necesario, la opción lógica pasa por modificar esa solución que en el pasado se aplicó, de manera que se intente obtener un resultado mejor.

Concepto de caso Para seguir entendiendo el razonamiento basado en casos, es necesario explicar el concepto de ‘caso’. Hasta ahora se ha hablado de razonamiento basado en casos y se ha tratado de explicar que se trata de un

razonamiento basado en la experiencia o conocimiento de casos previos. Esa experiencia o casos previos son precisamente lo que en este contexto significa caso. Un concepto que resulta fundamental en los sistemas CBR dado que su estructura se diseña sobre él. Un caso puede definirse como una parte de conocimiento contextualizado que representa una experiencia previa [34].

Mecanismo de razonamiento El modelo de actuación de un sistema CBR se conoce como ciclo de vida de un sistema CBR formado por cuatro procesos secuenciales [35]. Estos cuatro procesos, conocidos como *las cuatro 'R'*, son: *Retrieve* (Recuperación), *Reuse* (*Adaptación*), *Revise* (*Revisión*) y *Retain* (*Retención o aprendizaje*) [36].

Base o memoria de datos La memoria de casos es una de las partes más a tener en cuenta en un sistema CBR. En ella se tiene almacenado todo lo aprendido y se extraerán las soluciones anteriores. También en esta memoria se mantiene la representación y la organización de los casos, por lo que se debe tener en cuenta la estructura de los casos y tratar de facilitar, dentro de lo posible, todas las operaciones que se realizan en el ciclo del sistema CBR, a través de la indexación y organización de los casos.

Aplicaciones de sistemas CBR en otros estudios médicos A finales de los años 80, los sistemas CBR aparecieron como una alternativa interesante para construir aplicaciones médicas de inteligencia artificial y desde entonces está establecida en este campo [37]. Una de las características atractivas de los CBR en la medicina es que los conceptos de paciente y enfermedad se adaptan de manera natural a la arquitectura de estos sistemas. Aunque se han identificado varias ventajas de usar CBR en la medicina, el campo de la medicina no está exento de problemas. Alguno de los cuales, afectan específicamente a sistemas CBR.

En [38] se presentan las siguientes ventajas de la aplicación de sistemas CBR en la medicina:

Coincidencia de conceptos naturales: cognitivamente, como ya se ha mencionado, los sistemas CBR se asemejan a la forma en que los médicos razonan los casos de los pacientes y la forma en que utilizan los casos que se encuentran en su conocimiento.

Experiencia explícita: un sistema CBR se adapta de forma natural para ajustarse a los requisitos específicos de una cierta clínica o un cirujano.

La dualidad del conocimiento objetivo y subjetivo: en lugar de utilizar el conocimiento subjetivo de uno o más expertos para construir sistemas (como por ejemplo se hace para sistemas expertos basados en reglas), los sistemas CBR se basan en los casos existentes (que pueden o no ser completamente conocidos).

Adquisición automática de conocimiento subjetivo: los sistemas CBR exhiben una adquisición de conocimientos incrementales y el conocimiento se puede abstraer al generalizar los casos.

Integración de sistemas: la información de pacientes que se recoge ya en hospitales y por los médicos, se almacena en medios de lectura mecánica, lo que simplifica la integración con los sistemas CBR que pueden utilizarlos (siempre realizando las adaptaciones correspondientes).

Sin embargo, las desventajas de las que se hablaban previamente, en [38] se indica que son los siguientes:

Adaptación: debido a que resulta habitual que exista una cantidad extremadamente grande de características que intervienen en un caso médico, la adaptación de los casos se convierte un problemática. La generalización y una eficiente identificación de métodos ayuda a remediar en parte a esta cuestión, pero en general, este problema persiste según [39].

Falta de fiabilidad: aunque la fiabilidad de un sistema CBR aumenta con la proporción de la cobertura del problema, la fiabilidad no puede ser garantizada. La adición de nuevos casos no hace necesariamente que el sistema converja hacia una mayor fiabilidad.

Concentración en la referencia: los sistemas CBR se concentran en la referencia, por lo que no pueden funcionar como fuentes de experiencia anterior a menos que exista un caso adecuado en la memoria de casos.

2.3. Metodologías existentes

Existe una gran variedad de metodologías propuestas en diferentes trabajos que guardan relación con el que se describe. La mayoría, basados en el análisis de imágenes médicas a partir de diferentes técnicas empleadas. A continuación, se enumeran las técnicas más utilizadas con este propósito en el ámbito del análisis de imágenes del fondo de ojo y se realiza una comparativa entre las diferentes técnicas en base a su utilización en diferentes trabajos.

Computacionalmente, las primeras investigaciones que consideran el problema de la extracción de la vasculatura de la retina a partir de imágenes estaban basados en la utilización de los siguientes métodos [40]:

Filtrado adaptado: este método se ha utilizado para enfatizar los vasos en una imagen del fondo de ojo en múltiples trabajos, inicialmente propuesto en [41]. Puesto que los vasos son Gaussianos transversalmente, se utiliza generalmente un núcleo de convolución con una sección transversal de Gauss. Este método es muy eficaz, pero no suele ser útil su aplicación de manera individual. En [42], ya en el año 2001, se plantea un algoritmo que encaja en esta categoría y que identifica los vasos de manera individual, pero con intervención manual.

Métodos morfológicos: el procesamiento de imágenes de manera morfológica es muy eficaz en la extracción de características de imágenes cuya forma se conoce a priori. Para casos similares al de este trabajo, se ha empleado para obtener la segmentación o esqueletización de la vasculatura de forma exitosa en [43], [44] o [45].

Algoritmos de seguimientos de vasos: Estos algoritmos de seguimiento fueron por primera vez introducidos en [46] y se fundamentan en la segmentación de un vaso entre dos puntos. Este algoritmo se aplica a lo largo de todo el trayecto de cada vaso y se mide también el grosor. La principal ventaja de esta metodología de seguimiento de vasos es que ofrece el ancho de los vasos con gran precisión y puede proporcionar información acerca de diferentes zonas del vaso. La principal desventaja que tiene la aplicación de estos algoritmos es la necesidad de tener un punto de partida y, por lo general, de final.

Técnicas manuales: Con el fin de realizar mediciones precisas de la dimensión fractal, varios investigadores ([47], [48]) indicaron que, la metodología más fiable para realizar el análisis era un trazado manual de la imagen digital usando una tableta digitalizadora o incluso un calibre.

Más recientemente han aparecido todo tipo de trabajos teóricos y herramientas que aplican los modelos propuestos en la teoría sobre el tratamiento de las imágenes de la retina. A continuación se detallarán algunas de las más importantes:

En [49] se presenta un nuevo algoritmo de procesamiento de imagen automático para la detección de exudados en retinografías, esto es, un procesamiento muy similar al que se desea realizar, pero algo menos completo. No obstante, las técnicas aplicadas para el preprocesado, donde se trata de aumentar el contraste existente el objetivo a detectar y el fondo, frecuentemente pobre en las imágenes originales son similares y basadas en la obtención de una imagen en escala de grises a partir de las tres capas de colores (roja, verde y azul) en lugar de utilizar una única.

Para la medición de los vasos sanguíneos a la hora de analizar enfermedades como la diabetes, la hipertensión o la retinopatía prematura, en [50] se presentan diferentes técnicas. Su método de segmentación se compone de técnicas multiescala puesto que los vasos sanguíneos de la retina tienen una gama de diferentes tamaños.

En [51] se propone una combinación de una serie de técnicas para el tratamiento de imágenes de retinas con gran resolución en tres dimensiones. Se basan en métodos de tomografía de coherencia óptica (OCT por sus siglas en inglés *Optical Coherence Tomography*), que es una modalidad no invasiva de toma de imágenes en alta resolución a través de radiación óptica no ionizada.

Otro trabajo que se aproxima al propósito buscado es [52], realizado por el equipo médico involucrado en esta investigación y que da una primera aproximación a lo que se desea realizar de cara al análisis, pero quedando escaso en información extraída, por lo que se modificó parte de la metodología del trabajo realizado.

En [53] como resultado general, se realiza un análisis de numerosos trabajos existentes, donde se puede encontrar las diferentes técnicas aplicadas para el preprocesamiento, las técnicas aplicadas a la hora de localizar el disco óptico o papila y los algoritmos y filtros utilizados para la segmentación de la vasculatura.

3. Arquitectura propuesta

3.1. Introducción

A lo largo de este apartado, se detallará la metodología, las técnicas y la estructura propuesta para realizar análisis de imágenes de fondo de ojos, siendo capaz de extraer toda la información que el equipo clínico involucrado en la investigación cree necesario para continuar avanzando en el caso de estudio. Además se explicará la manera en la que se ha aplicado en la herramienta desarrollada y se analizará la forma en que se integrarán los resultados dentro de un sistema experto de razonamiento basado en casos para descubrir patrones y realizar diagnósticos de ciertas patologías.

3.2. Detalle de la arquitectura y metodología

La arquitectura que sigue su plataforma puede verse descrita en la Figura 1, quedando dividida en cinco grandes bloques secuenciales, representados con un círculo, que contienen a su vez una serie de subprocesos en los que se realizan una serie de tareas (en rectángulos).

La metodología utilizada para lograr la funcionalidad completa de la plataforma se puede dividir en dos partes diferenciadas:

La primera fase se ha denominado “extracción de la información” de la imagen de la retina. En esta fase se detectan e identifican las partes de la retina que se consideran relevantes y se traduce en la extracción de sus características a modo de medición etiquetada. Estas mediciones son suficientes como para representar las características de la imagen original sin presencia de la imagen como tal. Esta fase está supervisada por un experto que podrá realizar modificaciones de manera manual si el procedimiento automático que realiza el sistema no generase un resultado preciso.

En la segunda fase se realiza la denominada etapa de “diagnóstico”, que toma como entradas los datos generados en la fase previa de extracción de la información en conjunto con los datos ya extraídos y contrastados ya existentes en el sistema. Con estas entradas se produce la extracción de conocimiento.

Este subapartado se centra en la primera fase, encargada de crear e identificar todos los parámetros de interés de la retina. Para llevar a cabo esta etapa, se procesan los siguientes bloques:

Gestión de pacientes Este bloque hace referencia a la gestión de las variables clínicas, comunes a todos los pacientes, y a la gestión de los propios pacientes, donde las variables clínicas reciben ya un valor específico.

Es importante de cara al bloque de extracción de conocimiento asociar los resultados a las variables clínicas de cara a encontrar posibles relaciones entre ellos.

La herramienta permite insertar nuevas variables asociadas a pacientes acompañadas de una descripción. Después, tomarán un valor específico o no (según se necesite), una vez sea insertado un nuevo paciente en la base de datos.

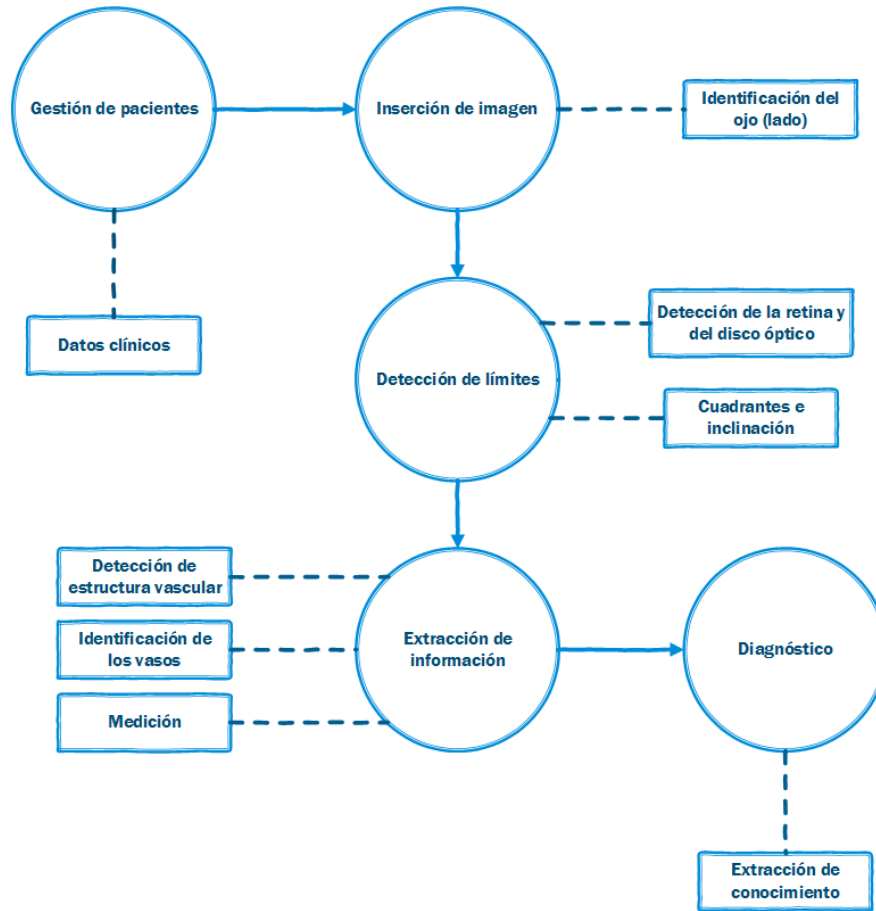


Figura 1. Esquema de la arquitectura propuesta

Los pacientes, para respetar la ley de protección de datos, precisan únicamente de un campo obligatorio, ID, que el usuario asigna de manera que lo identifique de manera única (puede ser un identificador interno o, en caso de no precisar de confidencialidad, el DNI). El resto de campos quedan únicamente para información interna a la hora de diferenciar los pacientes y, por el momento, no se tienen en cuenta a la hora del análisis.

La realización de esta inserción de pacientes es necesaria a la hora de poder procesar las imágenes que se desean analizar. Estas imágenes, una vez tratadas, quedan almacenadas de manera que se puedan revisar accediendo a la información del paciente.

Inserción de la imagen Este es el bloque más sencillo y el único de la parte de procesamiento que requiere de manera obligatoria una acción manual por

parte del supervisor. Ha de escoger la imagen que desea analizar una vez haya seleccionado previamente el paciente al que corresponde la imagen. Hecho esto, la herramienta procede con la detección de la mácula.

Inicialmente de aplicar ningún filtro, como parte del preprocesamiento, puesto que la mácula siempre estará centrada longitudinalmente en las imágenes a tratar, se reduce su área de detección a las dos partes centrales resultantes de dividir la imagen en 4 sub-partes del mismo tamaño, con el ancho total de la imagen y un cuarto de la altura. El resultado se puede ver en la Figura 2.

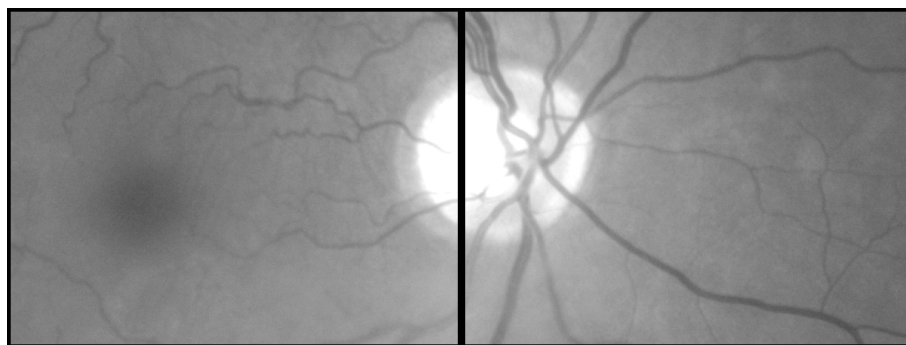


Figura 2. Preprocesamiento en busca de la posición de la mácula

El procedimiento de filtrado que se sigue en busca de la mácula se encarga de detectarla y devolver el lado asociado al ojo del que se ha tomado la imagen que se está analizando. Este procedimiento se compone de los siguientes pasos y filtrados:

- Extracción de una copia de la imagen original con un único canal de manera que se obtiene una nueva imagen en escala de grises. Para ello, primeramente se reserva memoria con la función *cvCreateImage* y, posteriormente, se almacena en ese espacio la copia únicamente con una combinación de cada uno de los canales (RGB) para mejorar el contraste, adaptando a las necesidades la técnica propuesta por [49].

- Se crean regiones de interés con las características de tamaño especificadas y se asocian las mitades izquierda y derecha a dos nuevas imágenes.

- Se obtiene la media de intensidad de los píxeles de cada una de las dos nuevas con la función *cvMean* y se tiene que la que mayor intensidad tenga (derecha o izquierda) es en la que se encuentra la mácula.

Pese a que puede parecer un procedimiento bastante sencillo, ha ofrecido un resultado del 100 % en la fase de pruebas (con 50 muestras), lo que no sucedió con otros métodos aplicados, como la binarización de la imagen con un umbral dinámico y analizando la cantidad de píxeles negros, tanto contabilizando el total de los presentes en la imagen, como únicamente agrupaciones de píxeles.

Detección de límites Este paso contiene diferentes módulos que se encargarán de localizar el disco óptico o papila, detectar los bordes (y el centro) de la retina y el disco óptico y posicionar el centro y la inclinación de los cuadrantes en los que se divide la retina. El principal objetivo es construir una estructura de datos basada en matrices de colores que identifique cada una de las partes de la retina a partir de la imagen original.

Para ello, como paso previo se realizan técnicas de filtrado para producir una reducción del ruido, incrementar el contraste, agudizar los bordes y corregir efectos borrosos. En concreto, en este paso se utilizan técnicas de procesamiento de imágenes basadas en bordes de la estructura, como en [54] y [55] para detectar su intensidad.

La detección del borde de la retina se realiza mediante un sencillo procedimiento de análisis de imágenes, puesto que en todas las imágenes utilizadas se colocan sobre un fondo negro o más oscuro.

Siguiendo esta regla general, la detección del borde de la retina se realiza mediante un sencillo procedimiento de análisis de imágenes, puesto que en todas las imágenes utilizadas se colocan sobre un fondo negro o más oscuro.

La detección de los bordes del disco óptico es un proceso que combina técnicas de filtrado de imagen con la Transformada de Hough [18]. Detección que mejora teniendo en cuenta las circunstancias comunes: tamaño aproximado del disco óptico respecto a la retina y ubicación aproximada. También supone una ayuda en la detección el hecho de conocer el tamaño aproximado del disco, que guarda una relación con el tamaño total de la retina de entre 1:4 y 1:6. Además, al conocer que estará en las proximidades del centro de la imagen, el análisis se centra únicamente en el cuadrado inscrito en el círculo asociado a los bordes de la retina.

La identificación de la papila marca el punto de partida desde donde se detectarán e identificarán los diferentes vasos sanguíneos. Además, su radio marca el área de detección que se tendrá en cuenta a la hora de realizar los siguientes pasos. Esta área de detección se define de manera concéntrica al disco óptico, con el triple del tamaño de éste.

El procedimiento de detección se compone de los siguientes pasos y filtrados:

- A partir de la imagen en escala de grises ya preprocesada, se crea la región de interés con las características de tamaño especificadas (cuadrado inscrito en los bordes de la retina).

- Se utilizan filtros de umbral dinámico, binarizando la imagen, zonas por su luminosidad que las discrimine para poder analizar únicamente las secciones más luminosas.

- Se aplican transformaciones para expandir y cerrar una serie de píxeles y se realiza una búsqueda de contornos (o *blobs*) para situaciones se hayan detectado otros puntos de luminosidad que cumplan con las restricciones del umbral establecido y que no se correspondan con la papila, algo que puede suceder puesto que las imágenes son muy cambiantes y se filtran por tamaño.

- Se ejecuta la transformada de Hough para búsqueda de círculos. El círculo de mayor tamaño que cumpla las características de la papila, será asociado como

la papila y su centro, como el centro de ésta, desde donde partirá el análisis de los vasos sanguíneos.

Se decidió que el radio de la papila debía establecerse por defecto para que el área analizada en cada imagen fuese siempre en proporción al tamaño total, por lo que una vez detectado el centro de la papila, el radio de ésta será un valor constante definido como la sexta parte del radio de la retina.

Extracción de la información Para realizar la detección de la estructura de los vasos sanguíneos se trabaja con la imagen original en escala de grises. Se busca inicialmente separar esta estructura del fondo para posteriormente procesarla, por lo que se puede pensar en resaltarla o en ‘suprimirla’ de manera selectiva para únicamente quedarse con el fondo. En este caso, se opta por la segunda técnica. Para realizar este filtrado, se le aplican al área de detección o de acción los siguientes pasos:

- La tonalidad de la papila se oscurece sustituyéndola por la tonalidad media del área de acción para evitar que al aplicar los filtros que se expliquen a continuación, la luminosidad propia de la papila se expanda por sus proximidades induciendo a error.

- Se aplican una serie de filtros gaussianos a la imagen en escala de grises generada en apartados anteriores, que hacen que se desenfoque la imagen. Primero se pasa 4 veces un filtro de la mediana y posteriormente se aplica 15 veces un filtro lineal.

- Una vez se obtiene la imagen resultante, se contrasta contra la imagen original en busca de cambios significativos (superiores a un umbral ajustado dinámicamente en tiempo de ejecución en función de los resultados) y todos aquellos píxeles en los que se produzcan esos cambios, serán candidatos a formar parte de la estructura vascular que se desea detectar.

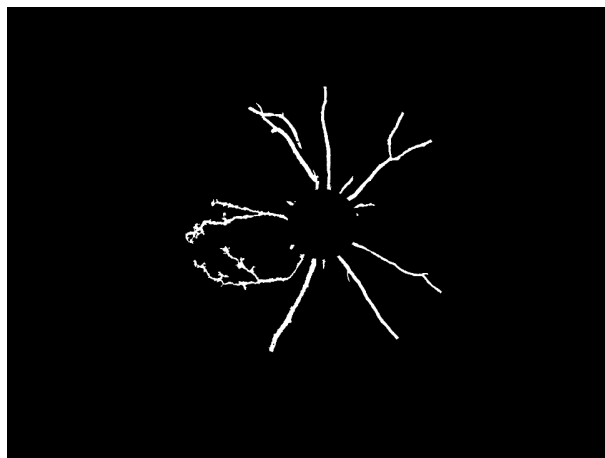


Figura 3. Resultado de la detección de la estructura de los vasos sanguíneos

- Para finalmente serlo, debe pertenecer a un ‘segmento’ que cumpla con una serie de características de longitud, grosor y área para discriminar el ruido que pueda existir. En caso de cumplirse, todos los píxeles del segmento pasan a formar parte de la estructura detectada. El resultado se puede ver en la Figura 3.

Con la estructura de vasos sanguíneos ya detectada, el siguiente paso consiste en la identificación de cada uno de ellos. Existen dos tipos de vasos sanguíneos: arterias y venas, cuya única diferencia perceptible a través de una imagen que las distinga en cada caso es la tonalidad. En todos los casos, las venas serán más oscuras que las arterias, pero esta tonalidad se ve además influenciada por la luminosidad en la zona de la imagen en la que esté el vaso.

Este proceso de identificación se basa en la implementación de un clasificador, cuyos valores se autoajustan siguiendo un algoritmo de balanceo diseñado para la problemática que se plantea en este trabajo.

Para analizar los cruces, que pueden suponer un problema al ocultar siempre el vaso superior al inferior, la primera metodología planteada pasa por la extracción del esqueleto de los vasos sanguíneos para clasificar sus puntos según su número de vecinos. Este esqueleto se extrae aplicando técnicas de K-medias, diagrama de Voronoy, triangulación de Delaunay y proceso degenerativo de rectificación, basándose en la metodología propuesta en [56], identificando como resultado 4 tipos de puntos: finales (con un vecino), normales (con dos vecinos), ramificaciones (3 vecinos) o intersecciones (4 o más vecinos).

El proceso de medición extrae más de 200 parámetros en función del tipo de vaso, su ubicación (distancia a la papila y cuadrante) sobre su grosor, longitud y área. Es importante para esta tarea tener en cuenta la escala del fabricante del retinógrafo, que nos dará una medida real para cada píxel de la foto y la corrección de la curvatura del ojo.

4. Resultados, conclusiones y líneas futuras de trabajo

Resultados Tras la aplicación de las metodologías y técnicas descritas en la herramienta que se ve en la Figura 4, ésta se utiliza para verificar que ese conjunto novedoso de procedimientos descrito es correcto. Para ello, un operador ha evaluado 100 imágenes de una submuestra aleatoria de 50 pacientes con una diferencia de una semana entre las dos realizaciones. En este caso, el operador (op1) y las imágenes analizadas fueron las mismas en los dos días y no tenía información de la evaluación previa realizada. Con esto se pretende demostrar que la herramienta es determinista al menos de una manera subjetiva.

Además, para demostrar la ausencia de la componente subjetiva en el análisis, se realizaron las mismas evaluaciones con un segundo operador (op2) diferente del primero. Ambos recibieron el mismo entrenamiento previo para conocer el funcionamiento de la herramienta.

Los resultados fueron muy parejos y el margen de error considerablemente pequeño, por lo que la componente subjetiva es pequeña y tampoco influye de



Figura 4. Resultado de la herramienta con los vasos sanguíneos detectados e identificados

manera apreciable la intervención del operador, como se puede ver a continuación:

Cuadro 1. Porcentajes de errores en las medidas

Parámetro	op1-op1	op1-op2
Longitud vena	1,742	1,731
Longitud arteria	1,62	1,42
Longitud vascular	3,362	3,151
Radio papila	0	0
AV Index	0,1973	0,224
Grosor vena	5,4919	5,519
Grosor arteria	5,5149	5,62
Grosor vascular	11,0068	11,139
Área vena	2,67	2,81
Área arteria	2,401	2,301
Área vascular	5,071	5,111

Además, se comparó el parámetro *AV index* con la medición verificada, donde únicamente se obtiene el parámetro del grosor de las arterias y las venas (cuyo cociente define el parámetro analizado), teniendo como resultado una diferencia de 0.012, que supone un 1.73 % (la media de valores del índice arteriovenoso, que carece de unidades, fue de 0.691).

En cuanto a tiempo de procesamiento, al compararlo con la herramienta que aplicaba las metodologías que previamente había desarrollado el equipo médico, con la que se procesa menor área y se generan menos parámetros, se obtuvo una mejora del 38.5 %.

Conclusiones La conclusión más importante, tras la validación de la herramienta auxiliar desarrollada para realizar mediciones, es que las metodologías y tecnologías puestas en práctica en base a las estudiadas y descritas a lo largo del presente documento, han sido correctas.

Gracias a esta validación se tiene una base sólida a la hora de evolucionar el trabajo realizando el sistema CBR propuesto, con el que se espera realizar diagnósticos fiables a partir de una serie de casos procesados y clasificados por un experto.

La solución propuesta permite, además, la utilización de las metodologías propuestas para el tratamiento y medición de cualquier imagen tomada con cualquier retinógrafo, lo que supone que la memoria de casos puede aumentar enormemente (siempre de manera segura y fiable) para suponer una mejor base de cara a la realización de diagnósticos clínicos.

Con la metodología planteada, que supone la combinación varias técnicas ya utilizadas en conjunción con las muchas técnicas y algoritmos desarrolladas para este trabajo y aplicados con éxito, se consigue obtener información específicamente solicitada por el equipo médico al creerla relevante para realizar la fase de integración con el sistema CBR.

El hecho de aplicar un conjunto de técnicas para cada paso detallado en la propuesta, en lugar de una única técnica como se ha detallado que realizan otras aplicaciones, se ha conseguido realizar en un tiempo de ejecución adecuado para el procesamiento de cada imagen.

Trabajo futuro Pese a que la metodología y las técnicas de análisis que se han diseñado han sido ya validadas, es necesario realizar algún ajuste que proponga una solución más precisa ante cruces entre diferentes vasos sanguíneos.

Una vez validada que esta metodología planteada es correcta y la medición es todavía más precisa y aplicable a retinas con situaciones de cruces múltiples entre diferentes vasos sanguíneos, se podrá crear la memoria de casos de una manera completamente fiable. Para ello, está pensado que un usuario supervisor experto procese una serie de imágenes (con 500 se cree que sería suficiente) correspondientes a pacientes con diferentes patologías y completamente sanos. Dichas imágenes pasarían a formar parte de la memoria de casos para el sistema CBR.

Gracias a esto, se podrán encontrar relaciones de estos resultados con nuevas patologías o factores que hacen que si toman ciertos valores, cierta parte de la población sea susceptible a padecer algún riesgo.

Bibliografía

- [1] Y. Tanabe, R. Kawasaki, J. J. Wang, T. Y. Wong, P. Mitchell, M. Daimon, T. Oizumi, T. Kato, S. Kawata, T. Kayama, *et al.*, “Retinal arteriolar narrowing predicts 5-year risk of hypertension in japanese people: the funagata study,” *Microcirculation*, vol. 17, no. 2, pp. 94–102, 2010.
- [2] G. Tikellis, D. K. Arnett, T. N. Skelton, H. W. Taylor, R. Klein, D. J. Couper, A. R. Sharrett, and T. Y. Wong, “Retinal arteriolar narrowing and left ventricular hypertrophy in african americans. the atherosclerosis risk in communities (aric) study,” *American journal of hypertension*, vol. 21, no. 3, pp. 352–359, 2008.
- [3] T. Y. Wong, B. B. Duncan, S. H. Golden, R. Klein, D. J. Couper, B. E. Klein, L. D. Hubbard, A. R. Sharrett, and M. I. Schmidt, “Associations between the metabolic syndrome and retinal microvascular signs: the atherosclerosis risk in communities study,” *Investigative ophthalmology & visual science*, vol. 45, no. 9, pp. 2949–2954, 2004.
- [4] H. Yatsuya, A. R. Folsom, T. Y. Wong, R. Klein, B. E. Klein, A. R. Sharrett, *et al.*, “Retinal microvascular abnormalities and risk of lacunar stroke atherosclerosis risk in communities study,” *Stroke*, vol. 41, no. 7, pp. 1349–1355, 2010.
- [5] K. McGeechan, G. Liew, P. Macaskill, L. Irwig, R. Klein, B. E. Klein, J. J. Wang, P. Mitchell, J. R. Vingerling, T. Paulus, *et al.*, “Meta-analysis: retinal vessel caliber and risk for coronary heart disease,” *Annals of internal medicine*, vol. 151, no. 6, pp. 404–413, 2009.
- [6] T. Y. Wong, R. Klein, A. R. Sharrett, B. B. Duncan, D. J. Couper, J. M. Tielsch, B. E. Klein, and L. D. Hubbard, “Retinal arteriolar narrowing and risk of coronary heart disease in men and women: the atherosclerosis risk in communities study,” *Jama*, vol. 287, no. 9, pp. 1153–1159, 2002.
- [7] C. Sabanayagam, A. Shankar, B. E. Klein, K. E. Lee, P. Muntner, F. J. Nieto, M. Y. Tsai, K. J. Cruickshanks, C. R. Schubert, P. C. Brazy, *et al.*, “Bidirectional association of retinal vessel diameters and estimated gfr decline: the beaver dam ckd study,” *American Journal of Kidney Diseases*, vol. 57, no. 5, pp. 682–691, 2011.
- [8] C. Sabanayagam, E. S. Tai, A. Shankar, J. Lee, C. Sun, and T. Y. Wong, “Retinal arteriolar narrowing increases the likelihood of chronic kidney disease in hypertension,” *Journal of hypertension*, vol. 27, no. 11, pp. 2209–2217, 2009.
- [9] M. van Hecke, J. Dekker, G. Nijpels, R. Stolk, R. Henry, R. Heine, L. Bouter, C. Stehouwer, and B. Polak, “Are retinal microvascular abnormalities associated with large artery endothelial dysfunction and intima-media thickness? the hoorn study,” *Clinical science (London, England: 1979)*, vol. 110, no. 5, pp. 597–604, 2006.
- [10] C. Cuspidi, S. Meani, M. Salerno, V. Fusi, B. Severgnini, C. Valerio, E. Catinini, A. Esposito, F. Magrini, and A. Zanchetti, “Retinal microvascular changes

- and target organ damage in untreated essential hypertensives,” *Journal of hypertension*, vol. 22, no. 11, pp. 2095–2102, 2004.
- [11] M. Masaidi, C. Cuspidi, V. Giudici, F. Negri, C. Sala, A. Zanchetti, G. Grassi, and G. Mancina, “Is retinal arteriolar–venular ratio associated with cardiac and extracardiac organ damage in essential hypertension?,” *Journal of hypertension*, vol. 27, no. 6, pp. 1277–1283, 2009.
- [12] R. Klette, *Concise computer vision*. Springer, 2014.
- [13] L. Barghout and J. Sheynin, “Real-world scene perception and perceptual organization: Lessons from computer vision,” *Journal of Vision*, vol. 13, no. 9, pp. 709–709, 2013.
- [14] D. Mery, “Visión artificial,” *Praca dyplomowa, Universidad de Santiago, Chile*, 2002.
- [15] S. Milan, H. Vaclav, and B. Roger, “Image processing, analysis, and machine vision,” *Photocopy Edition, Posts & Telecom Press, Peking*, 2002.
- [16] E. R. Davies, *Machine vision: theory, algorithms, practicalities*. Elsevier, 2004.
- [17] V. I. Kober, M. G. Mozerov, J. Alvarez-Borrego, and I. A. Ovseyevich, “Unsharp masking by the rank-order filters with spatially adaptive neighborhoods,” *PATTERN RECOGNITION AND IMAGE ANALYSIS C/C OF RASPOZNAVANIYE OBRAZOV I ANALIZ IZOBRAZHENII*, vol. 12, no. 1, pp. 46–56, 2002.
- [18] W. Chen, M. J. Er, and S. Wu, “Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 2, pp. 458–466, 2006.
- [19] K. Delac, M. Grgic, and T. Kos, “Sub-image homomorphic filtering technique for improving facial identification under difficult illumination conditions,” in *International Conference on Systems, Signals and Image Processing*, vol. 1, pp. 21–23, 2006.
- [20] S. Messelodi, C. M. Modena, N. Segata, and M. Zanin, “A kalman filter based background updating algorithm robust to sharp illumination changes,” in *Image Analysis and Processing–ICIAP 2005*, pp. 163–170, Springer, 2005.
- [21] B. M. Ege, O. K. Hejlesen, O. V. Larsen, K. Møller, B. Jennings, D. Kerr, and D. A. Cavan, “Screening for diabetic retinopathy using computer based image analysis and statistical classification,” *Computer methods and programs in biomedicine*, vol. 62, no. 3, pp. 165–175, 2000.
- [22] S. Kido, J. Ikezoe, H. Naito, J. Arisawa, S. Tamura, T. Kozuka, W. Ito, K. Shimura, and H. Kato, “Clinical evaluation of pulmonary nodules with single-exposure dual-energy subtraction chest radiography with an iterative noise-reduction algorithm,” *Radiology*, vol. 194, no. 2, pp. 407–412, 1995.
- [23] G. Deng and L. Cahill, “An adaptive gaussian filter for noise reduction and edge detection,” in *Nuclear Science Symposium and Medical Imaging Conference, 1993., 1993 IEEE Conference Record.*, pp. 1615–1619, IEEE, 1993.
- [24] D. Van De Ville, M. Nachtegael, D. Van der Weken, E. E. Kerre, W. Philips, and I. Lemahieu, “Noise reduction by fuzzy image filtering,” *Fuzzy Systems, IEEE Transactions on*, vol. 11, no. 4, pp. 429–436, 2003.

- [25] H. Ying, M. Zhang, and J.-C. Liu, "Fractal-based automatic localization and segmentation of optic disc in retinal images," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pp. 4139–4141, IEEE, 2007.
- [26] J. B. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundam. Inform.*, vol. 41, no. 1-2, pp. 187–228, 2000.
- [27] V. Grau, A. Mewes, M. Alcaniz, R. Kikinis, and S. K. Warfield, "Improved watershed transform for medical image segmentation using prior information," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 4, pp. 447–458, 2004.
- [28] X. Jiang and D. Mojon, "Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 1, pp. 131–137, 2003.
- [29] A. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response," *Medical Imaging, IEEE Transactions on*, vol. 19, no. 3, pp. 203–210, 2000.
- [30] H. Rowley, S. Baluja, T. Kanade, *et al.*, "Neural network-based face detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 1, pp. 23–38, 1998.
- [31] G. Gardner, D. Keating, T. Williamson, and A. Elliott, "Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool.," *British journal of Ophthalmology*, vol. 80, no. 11, pp. 940–944, 1996.
- [32] P. Mills, "Efficient statistical classification of satellite measurements," *International Journal of Remote Sensing*, vol. 32, no. 21, pp. 6109–6132, 2011.
- [33] R. L. De Mantaras and E. Plaza, "Case-based reasoning: an overview," *Ai-communications*, vol. 10, no. 1, pp. 21–29, 1997.
- [34] R. C. Schank, "Dynamic memory: a theory of learning in computer and people," *Cambridge University, Press, New York*, 1982.
- [35] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI communications*, vol. 7, no. 1, pp. 39–59, 1994.
- [36] R. Lopez De Mantaras, D. McSherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M. L. Maher, M. T. Cox, K. Forbus, *et al.*, "Retrieval, reuse, revision and retention in case-based reasoning," *The Knowledge Engineering Review*, vol. 20, no. 03, pp. 215–240, 2005.
- [37] M. Nilsson and M. Sollenborn, "Advancements and trends in medical case-based reasoning: An overview of systems and system development.," in *FLAIRS Conference*, pp. 178–183, 2004.
- [38] L. Gierl, M. Bull, and R. Schmidt, "Cbr in medicine," in *Case-Based Reasoning Technology*, pp. 273–297, Springer, 1998.
- [39] R. Schmidt and L. Gierl, "Case-based reasoning for medical knowledge-based systems," *Studies in health technology and informatics*, pp. 720–725, 2000.

- [40] B. M. Ege, O. K. Hejlesen, O. V. Larsen, K. Møller, B. Jennings, D. Kerr, and D. A. Cavan, "Screening for diabetic retinopathy using computer based image analysis and statistical classification," *Computer methods and programs in biomedicine*, vol. 62, no. 3, pp. 165–175, 2000.
- [41] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum, "Detection of blood vessels in retinal images using two-dimensional matched filters," *IEEE Transactions on medical imaging*, vol. 8, no. 3, pp. 263–269, 1989.
- [42] N. Chapman, N. Witt, X. Gao, A. Bharath, A. Stanton, S. Thom, and A. Hughes, "Computer algorithms for the automated measurement of retinal arteriolar diameters," *British Journal of Ophthalmology*, vol. 85, no. 1, pp. 74–79, 2001.
- [43] F. Zana and J.-C. Klein, "Robust segmentation of vessels from retinal angiography," in *Digital Signal Processing Proceedings, 1997. DSP 97., 1997 13th International Conference on*, vol. 2, pp. 1087–1090, IEEE, 1997.
- [44] F. Zana and J.-C. Klein, "A multimodal registration algorithm of eye fundus images using vessels detection and hough transform," *Medical Imaging, IEEE Transactions on*, vol. 18, no. 5, pp. 419–428, 1999.
- [45] G. K. Matsopoulos, N. Mouravliansky, K. K. Delibasis, K. S. Nikita, *et al.*, "Automatic retinal image registration scheme using global optimization techniques," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 3, no. 1, pp. 47–60, 1999.
- [46] Y. Sun, "Automated identification of vessel contours in coronary arteriograms by an adaptive tracking algorithm," *Medical Imaging, IEEE Transactions on*, vol. 8, no. 1, pp. 78–88, 1989.
- [47] A. Daxer, "The fractal geometry of proliferative diabetic retinopathy: implications for the diagnosis and the process of retinal vasculogenesis," *Current eye research*, vol. 12, no. 12, pp. 1103–1109, 1993.
- [48] M. A. Mainster, "The fractal properties of retinal vessels: embryological and clinical implications," *Eye*, vol. 4, no. 1, pp. 235–241, 1990.
- [49] C. I. Sánchez, R. Hornero, M. I. López, M. Aboy, J. Poza, and D. Abásolo, "A novel automatic image processing algorithm for detection of hard exudates based on retinal image analysis," *Medical Engineering & Physics*, vol. 30, no. 3, pp. 350–357, 2008.
- [50] M. E. Martinez-Perez, A. D. Hughes, S. A. Thom, A. A. Bharath, and K. H. Parker, "Segmentation of blood vessels from red-free and fluorescein retinal images," *Medical image analysis*, vol. 11, no. 1, pp. 47–61, 2007.
- [51] A. G. Podoleanu and R. B. Rosen, "Combinations of techniques in imaging the retina with high resolution," *Progress in retinal and eye research*, vol. 27, no. 4, pp. 464–499, 2008.
- [52] L. García-Ortiz, J. I. Recio-Rodríguez, J. Parra-Sanchez, L. J. G. Elena, M. C. Patino-Alonso, C. Agudo-Conde, E. Rodríguez-Sánchez, M. A. Gómez-Marcos, *et al.*, "A new tool to assess retinal vessel caliber. reliability and validity of measures and their relationship with cardiovascular risk," *Journal of hypertension*, vol. 30, no. 4, pp. 770–777, 2012.
- [53] R. J. Winder, P. J. Morrow, I. N. McRitchie, J. Bailie, and P. M. Hart, "Algorithms for digital image processing in diabetic retinopathy," *Computerized Medical Imaging and Graphics*, vol. 33, no. 8, pp. 608–622, 2009.

- [54] C. Heneghan, J. Flynn, M. O’Keefe, and M. Cahill, “Characterization of changes in blood vessel width and tortuosity in retinopathy of prematurity using image analysis,” *Medical Image Analysis*, vol. 6, no. 4, pp. 407–429, 2002.
- [55] B. Chen, C. Tosha, M. Gorin, and S. Nusinowitz, “Analysis of autofluorescent retinal images and measurement of atrophic lesion growth in stargardt disease,” *Experimental eye research*, vol. 91, no. 2, pp. 143–152, 2010.
- [56] J. Mena, “Vectorización automática de una imagen binaria mediante k-means y degeneración de la triangulación de delaunay,” *Revista de la Asociación Española de Teledetección*, vol. 7, pp. 21–29, 2002.

Autores



Pablo Chamoso Santos es Ingeniero Técnico en Informática de Sistemas (2010) e Ingeniero Informático (2013) por la Universidad de Salamanca, donde también ha realizado el Máster en Comercio Electrónico (2010) y posteriormente el Master en Sistemas Inteligentes. Ha sido galardonado con distintos premios, como por ejemplo “Beca T-CUE”, con la que registró una propiedad intelectual en 2012 o el “Premio i3 a mejor proyecto fin de carrera o trabajo fin de Máster 2012-2014” concedido por el Colegio Profesional de Ingenieros en Informática de Castilla y León. Ha publicado diversos artículos relacionados sobre sus proyectos fin de carrera, trabajos fin de máster y sobre los diferentes proyectos de investigación en los que ha participado.

Emilio S. Corchado Rodríguez es profesor Titular acreditado como Catedrático de Universidad por la ANECA (Agencia Nacional de Evaluación de la Calidad y Acreditación) desde febrero de 2012. Ha sido elegido presidente del IEEE España durante el periodo 2014-2015. Es coautor de diversas patentes y propiedades intelectuales. Cuenta con una dilatada experiencia en la revisión y evaluación de proyectos de la Comisión Europea desde 2007. Es coautor de más de 60 publicaciones en revistas científicas con alto índice de impacto y también actúa como miembro del Consejo de Redacción de varias revistas de reconocido prestigio internacional. Ha sido invitado como editor en varias “Special Issues” de alto renombre. Es el organizador de diversos congresos científicos como (HAIS, IDEAL, SOCO, CISIS, NABIC, CASON, NWESP, WICT). Ha participado en varios proyectos Europeos (IT4Innovation, APP for AAL, ENGINE, SOFTCOMP, EMaCS, IntelliCIS, NISIS). Además ha sido Investigador Principal de diversos proyectos a nivel nacional y regional, también ha estado trabajando en estrecha colaboración con empresas como Grupo Antolín, Inmotia, Infranor, etc. Sus líneas de investigación se centran en el estudio de las redes neuronales, con especial énfasis en la Búsqueda Exploratoria de Proyecciones, Maximum Likelihood Hebbian Learning, Mapas Auto-Organizados, Redes SOM, Sistemas de Clasificación Múltiple y Sistemas Híbridos. Además de la optimización de procesos industriales y energéticos.





Juan M. Corchado Rodríguez es catedrático de Ciencia de la Computación e Inteligencia Artificial. Actualmente es Vicerrector de Investigación y Transferencia de la Universidad de Salamanca y Director del Grupo en Biotecnología, Sistemas Inteligentes y Tecnología Educativa de esta misma universidad (<http://bisite.usal.es>). Doctor en Informática por la Universidad de Salamanca, España (1998) y doctor en Inteligencia Artificial por la University of the West of Scotland, Reino Unido (2000). Es Coordinador Académico del Instituto de Arte y Tecnología de la Animación. Anteriormente ha sido Subdirector de la Escuela Superior de Ingeniería Informática en la Universidad de Vigo (España, 1999-00) e investigador en la Universidad de Paisley (1995-98). Colabora, como investigador, con el Laboratorio Oceanográfico de Plymouth (Reino Unido) desde 1993. Ha dirigido más de 70 proyectos de Inteligencia Artificial con financiación tanto internacional, como nacional o autonómica. Ha dirigido 16 tesis doctorales y es co-autor de más de 300 libros, capítulos de libros, artículos en revistas científicas, etc. la mayoría de ellos presentan aspectos tanto prácticos como teóricos relacionados con los Sistemas Híbridos de Inteligencia Artificial, la Biomedicina, la Inteligencia Ambiental, los Sistemas Inalámbricos y la Oceanografía. Es autor de más de una veintena de propiedades intelectuales y patentes.

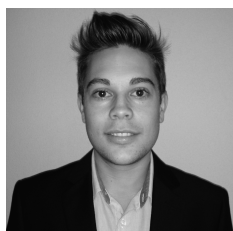
Francisco Javier Esteban Vicente es Ingeniero Técnico en Informática de Sistemas (2012) y Graduado en Ingeniería Informática (2013) por la Universidad de Salamanca. En la misma Universidad también ha realizado el Master de Sistemas Inteligentes, especializándose en reconocimiento facial y robótica. Actualmente ha sido admitido en la prestigiosa Universidad Técnica de Munich (Technische Universität München, TUM) donde complementará sus estudios estudiando un Máster en Computación Biomedica. Sus líneas de investigación se centran en los siguientes campos: Bioinformática, Robótica y Redes Neuronales.





Francisco José García Peñalvo es Ingeniero Informático por la Universidad de Valladolid y Doctor en Informática por la Universidad de Salamanca. Actualmente desarrolla su actividad académica como Profesor Titular del Departamento de Informática y Automática de la Universidad de Salamanca. Desde el punto de vista de investigación dirige el GRupo de investigación en InterAcción y eLearning (GRIAL) de esta Universidad, contando entre sus principales líneas de investigación la Interacción Persona-Ordenador, el *eLearning*, la Ingeniería Web, la Arquitectura del *Software*, la Informática Educativa, y la Teoría de la Comunicación. Es autor de más de un centenar de artículos en revistas y congresos internacionales y ha participado en más de cincuenta proyectos de investigación de carácter regional, nacional, europeo e internacional en los últimos diez años. Por último, en relación con la actividad de gestión, fue Vicedecano de Innovación y Nuevas Tecnologías de la Facultad de Ciencias de enero de 2004 a marzo de 2007 y Vicerrector de Innovación Tecnológica de la Universidad de Salamanca de marzo de 2007 a diciembre de 2009. Fue promotor y presidente del Comité de Dirección del Observatorio de la actividad, la innovación y las tendencias en la Formación en Red-SCOPEO desde su creación hasta diciembre de 2009.

Angélica González Arrieta es doctora en Informática por la Universidad de Salamanca. Cuenta con una amplia experiencia investigadora en el campo de la Computación Neuroborrosa y las Redes Neuronales. Es Profesora Titular del Departamento de Informática y Automática de dicha Universidad. Actualmente compatibiliza su labor docente e investigadora con la dirección de diversas actividades formativas sobre seguridad informática, colaborando activamente con la Academia de la Policía Nacional de Ávila.



Alfonso González Briones natural de Salamanca. Es Ingeniero Técnico en Informática de Sistemas (2012) y Graduado en Ingeniería Informática (2013) por la Universidad de Salamanca, donde también ha cursado el Máster en Sistemas Inteligentes especializándose en *Computer Vision*. Actualmente forma parte del grupo de investigación BISITE (Biomedicina, Sistemas Inteligentes y Tecnología Educativa), también en la Universidad de Salamanca, donde realiza su tesis doctoral. Sus líneas de investigación se centran en la Inteligencia Artificial, Minería de datos y el Reconocimiento de Patrones, con especial énfasis en la aplicación de estas técnicas en el campo de la Bioinformática.

Antonio González Torres obtuvo el Grado en Informática Empresarial y el Magister en Computación e Informática en la Universidad de Costa Rica. Posteriormente, realizó los estudios para obtener el Diploma de Estudios Avanzados del Doctorado en Informática y Automática y concluyó el Master en Sistemas Inteligentes de la Universidad de Salamanca. Además, efectuó una estancia de investigación en la Open University del Reino Unido. Es instructor certificado por Cisco Systems (para CCNA y CCNA Security) y cuenta con varias certificaciones profesionales en sistemas de Microsoft. Durante su trayectoria profesional, se ha desempeñado como informático en el sector público y privado de Costa Rica (Intel Corporation, Sykes Corporation y el Museo Nacional de Costa Rica), así como en varias empresas españolas (ENIAC Formación Salamanca y Grupo Global Exchange). Ha sido profesor de grado y posgrado en la Universidad de Costa Rica (UCR) y la Universidad Latina de Costa Rica (ULATINA). Actualmente se desempeña como profesor investigador en la Universidad Latinoamericana de Ciencia y Tecnología (ULACIT) y como Analista de Seguridad en el Threat Management Center de Equifax, Inc.

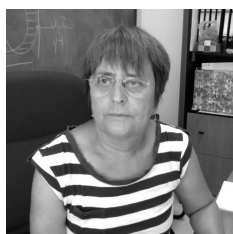


Óscar Gonzalez Velasco natural de Salamanca, cursó estudios de Ingeniería Técnica en Informática por la Universidad de Salamanca, posteriormente cursó estudios de Ingeniería Superior en Informática en la misma universidad, periodo en el cual realizó un intercambio con la University of the West of Scotland, Reino Unido, mediante el programa Erasmus obteniendo el título de Bachelor of Science in Computing Sciences (Honours class). Seguidamente realizó el Master en Sistemas Inteligentes de la Universidad de Salamanca, especializándose en el campo de la bioinformática.

Alberto López Barriuso natural de Salamanca, es Ingeniero Técnico en Informática de Gestión (2010) por la Universidad Pontificia de Salamanca y Graduado en Ingeniería Informática (2011) por la Universidad Pontificia de Salamanca, año en el que realizó un intercambio en la Univerza v Mariboru (Eslovenia) a través del programa Erasmus. Posteriormente realizó el Máster en Sistemas Inteligentes en la Universidad de Salamanca. Durante sus estudios ha formado parte del Club Universitario de Innovación de la Universidad Pontificia de Salamanca, participando como co-autor, diseñador y desarrollador en el proyecto Aescolapius, destinado al control de una silla de ruedas motorizada mediante diferentes técnicas. Actualmente trabaja como investigador y

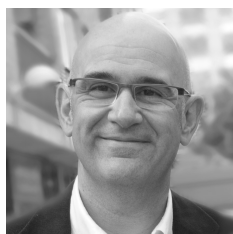


desarrollador en el Grupo de Investigación BISITE (Biomedicina, Sistemas Inteligentes y Tecnología Educativa), en el que ha centrado su trabajo en las áreas de Agentes Inteligentes y Sistemas Multiagente.



María Gracia Manzano Arjona es catedrática de Lógica y Filosofía de la Ciencia en la Universidad de Salamanca. Doctora en Filosofía por la Universidad de Barcelona obtuvo una beca postdoctoral Fullbright en Berkeley, California University, Estados Unidos en 1978 cuyo director fue Leon Henkin. Visiting Professor en University of Swansea (Reino Unido, 1991) y en Stanford University (Estados Unidos, 1996). Ha realizado estancias en los siguientes centros extranjeros The Center for Theoretical Computer Science, University of Leeds (Leeds, Reino Unido, 1986), Department of Mathematics and Computer Science, Queen Mary College (Londres, Reino Unido, 1985), Mathematical Institute of the Hungarian Academy of Science (Budapest, Hungría, 1992), Langue et Dialogue, Universidad de Nancy (Nancy, Francia, 2008) y en la Universidad Nacional Autónoma de Mexico (UNAM) en 2009. Es autora y co-autora de numerosos libros, capítulos de libros y artículos en revistas científicas y organizadora de congresos en el ámbito de la Lógica Híbrida, Lógica Intensional, Teoría de tipos, Aplicaciones de la lógica, Lógicas para la red, Herramientas para la enseñanza de la lógica: implementación informática, Teoría de Modelos, Lógicas modales, temporales y dinámicas, Lógicas de orden superior, Lógica heterogénea, Historia de la lógica, Lógica con diagramas y juegos lógicos.

Isabel Martín Rodríguez nació en Salamanca en el año 1991. Estudió Ingeniería Técnica en Informática de Sistemas (2012) y Grado en Ingeniería Informática (2013) en la Universidad de Salamanca, obteniendo de manera paralela el Grado Profesional en el Conservatorio de Salamanca. Posteriormente realizó el Máster en Sistemas Inteligentes de la misma universidad centrando sus líneas de investigación en torno a los sistemas inteligentes.



Luis Antonio Miguel Quintales Doctor en Informática y Automática por la Universidad de Salamanca. Es profesor titular en el Departamento de Informática y Automática de la Universidad de Salamanca e investigador en el Instituto de Biología Funcional y Genómica (IBFG) centro mixto de investigación de titularidad compartida entre el Consejo Superior de Investigaciones Científicas (CSIC)

y la Universidad de Salamanca. Las líneas de investigación en las que trabaja son la computación de altas prestaciones, la minería de datos y la genómica computacional. Como resultado de su trabajo de investigación ha publicado en algunas de las revistas más prestigiosas del ámbito de la Bioinformática como son: Briefings in Bioinformatics, Bioinformatics y BMC Bioinformatics.

Rodrigo Olmo Rodríguez es graduado en Ingeniería Informática por la Universidad Pontificia de Salamanca y Máster en Sistemas Inteligentes por la Universidad de Salamanca. Durante sus estudios de grado formó parte de dos proyectos del Club Universitario de Innovación: City-Bugger (plataforma web y aplicación móvil que permite identificar lugares con barreras de accesibilidad) y Proyecto MAR (asistente virtual de rehabilitación para personas con lesiones físicas). Actualmente desarrolla su actividad profesional como emprendedor de una startup tecnológica.



Juan Francisco de Paz Santana Doctor en Informática y Automática por la Universidad de Salamanca en 2010. Actualmente es Profesor Ayudante Doctor en Departamento de Informática y Automática de la Universidad de Salamanca e investigador del grupo de BISITE (<http://bisite.usal.es>). Ingeniero técnico en Informática de Sistemas (2003), Ingeniero Informático (2005) y Diplomado en Estadística (2007) por la Universidad de Salamanca. Es coautor de 130 artículos publicados en diversas revistas de ámbito nacional e internacional, workshops y simposios. Es miembro del Grupo de Investigación BISITE de la Universidad de Salamanca, donde ha participado en diversos proyectos de I+D.

Victor Parra Vidales natural de Zamora, es Ingeniero Técnico en Informática de Sistemas (2010) e Ingeniero Informático (2013), cuyo proyecto le proporcionó una beca en el grupo de investigación BISITE. Ha completado su formación con el Máster Oficial en Sistemas Inteligentes, impartido en la Universidad de Salamanca. Ha sido galardonado con el premio “Beca T-CUE”, el cual le permitió registrar una propiedad intelectual en el año 2013. También ha realizado algunas publicaciones sobre sus proyectos y trabajos de investigación. Actualmente forma parte del grupo BISITE, llevando a cabo labores de investigación y desarrollo de distintos proyectos informáticos.





Belén Perez Lancho es Profesora Titular del Área de Ingeniería de Sistemas y Automática. Es licenciada en Física y Doctora en Ciencias por la Universidad de Salamanca y realizó su formación posdoctoral en la Universidad Pierre et Marie Curie de París en el curso 1996-97. Imparte docencia en las titulaciones de Grado en Informática, Grado en Física y en el Máster en Sistemas Inteligentes. Su principal área de investigación son los sistemas inteligentes aplicados a la simulación y al control de procesos. Ha codirigido dos tesis doctorales y numerosos Proyectos de Fin de Carrera. En gestión académica ha ocupado el cargo de Secretaria Académica de la Facultad de Ciencias (2000-2004) y desde 2008 hasta la actualidad es Vicedecana de Infraestructuras e Innovación, presidiendo o participando en diferentes comisiones (Planes de estudios de Grado, Proyectos Fin de Carrera, Calidad, etc.).

Sara Rodríguez González es Profesor Ayudante Doctor en el Departamento de Informática y Automática la Universidad de Salamanca. Obtuvo su doctorado en Informática en la misma Universidad en 2010. Recibió el título de Ingeniería Técnica en Informática de Sistemas en 2004 y de Ingeniería en Informática en 2007. Por otra parte, ha llevado a cabo otros estudios de postgrado como el Master en el desarrollo de sistemas de comercio electrónico y el Master en animación de digital. Ha participado como co-autora en artículos publicados en revistas internacionales de reconocido prestigio como Applied Soft Computing, International Journal of Mathematical Imaging and Vision o Expert System with Applications y ha publicado otros más de 40 artículos en una gama de ámbitos que van desde los agentes y sistemas multiagente a la aplicación de sistemas en la bioinformática o la animación digital. También ha participado como comité de programa en conferencias como PAAMS, IWANN, PACBB, ISAMI, HAIS o DCAI. En la actualidad, es miembro de la Asociación Española para la Inteligencia Artificial (AEPIA), del Instituto Universitario de Investigación de Arte y Tecnología de la Animación (ATA) de la Universidad de Salamanca y del grupo de investigación BISITE (Bioinformática, Sistemas Inteligentes y Tecnología) en el que ejerce su labor investigadora.



Antonio Juan Sánchez Martín es Ingeniero Informático por la Universidad de Salamanca y graduado en Diseño Grafico en el Instituto Superior de Enseñanzas Artísticas de la Comunidad Valenciana. Sus focos de investigación se centran en cloud compiling, computer graphics y el desarrollo web. Participa en diferentes proyectos de

investigación a nivel nacional, ha publicado artículos en conferencias internacionales de prestigio, también tiene experiencia en empresas nacionales, trabajando en el desarrollo de proyectos web y gráficos por ordenador.

Roberto Therón Sánchez cursó sus estudios de Informática en la Universidad de Salamanca (Diplomatura) y la Universidad de la Coruña (Licenciatura). Tras entrar a formar parte del Grupo de Investigación en Robótica de la Universidad de Salamanca, presentó su trabajo de Tesis recibiendo el Premio Extraordinario de Doctorado. Posteriormente ha obtenido los títulos de Licenciado en Comunicación Audiovisual (Universidad de Salamanca) y Licenciado en Humanidades (Universidad de Salamanca). En la misma Universidad de Salamanca continúa realizando su trabajo de investigador, como encargado del grupo VisUsal (dentro del Grupo de Investigación Reconocido MiDa) que se centra en la combinación de enfoques procedentes de la Informática, Estadística, Diseño Gráfico y Visualización de Información, para obtener una adecuada comprensión de conjuntos de datos complejos. En los últimos años, se ha dedicado al desarrollo de herramientas de visualización avanzada para datos multidimensionales. En el área de Analítica Visual desarrolla productivas colaboraciones con grupos e instituciones de reconocido prestigio internacional, como el Laboratorio de Ciencias del Clima y del Medio Ambiente (París) o el Centro de Analítica Visual Avanzada de la ONU (Suiza). Es autor de más de 70 artículos en revistas y congresos internacionales.



Juan Carlos Uclés Mármol Cursó sus estudios de Ingeniería en Computación en la Universidad Tecnológica de Honduras. Desde 2008 hasta 2010 formó parte de la Dirección Ejecutiva de Ingresos (DEI), Institución Gubernamental enfocada a recolectar los impuestos tributarios y aduaneros de Honduras, después se desempeñó como encargado de medios digitales y publicidad para el Programa Presidencial de Educación Primaria e Integración Tecnológica (Educatrachos) en Honduras. Aquí mismo presentó su Tesis de Grado, “Control de Calidad en el Software”. Posteriormente obtuvo una beca para cursar sus estudios de Maestría obteniendo el título, “Máster en Sistemas Inteligentes” en la Universidad de Salamanca. Actualmente reside en la ciudad de Fort Lauderdale, Florida, Estados Unidos de Norte América, continuando sus estudios como complemento al área de Ingeniería en Gestión de Proyectos. Se ha especializado en Marketing Digital, Desarrollo Web, E-commerce, E-learning y Análisis de Redes Sociales.

El Máster Oficial en Sistemas Inteligentes de la Universidad de Salamanca tiene como principal objetivo promover la iniciación de los estudiantes en el ámbito de la investigación. El congreso organizado por el Departamento de Informática y Automática proporciona la oportunidad ideal para que sus estudiantes presenten los principales resultados de sus Trabajos de Fin de Máster y obtengan una realimentación del interés de los mismos.

Editores:

Alfonso González Briones

Isabel Martín Rodríguez

ISBN-10 84-697-1520-8

ISBN-13 978-84-697-1520-8

Departamento de Informática y Automática
Facultad de Ciencias, Universidad de Salamanca
<http://informatica.usal.es>