

Selección inteligente de un sistema de gestión de contenidos

Diego Corsi

Universidad Tecnológica Nacional, Facultad Regional Buenos Aires
Medrano 951, (C1179AAQ) Ciudad Autónoma de Buenos Aires, Argentina
diegocorsi@gmail.com

Recibido el 12 de Abril del 2010, aprobado el 23 de Abril de 2010

Resumen

En este trabajo se presenta un sistema inteligente basado en mapas de Kohonen (SOM) que puede ayudar a seleccionar - de entre un grupo predeterminado de sistemas de gestión de contenidos o CMS (Content Management Systems) - aquel que mejor cumpla con una serie de requerimientos.

Su desarrollo fue motivado por la dificultad de aplicar metodologías de selección convencionales, pues los CMS disponibles en el mercado son demasiado numerosos y diversos.

En los mapas generados se identifican con facilidad los CMS que más se aproximan a las características requeridas y, a partir de allí, pueden usarse métodos convencionales de evaluación.

PALABRAS CLAVES: SISTEMAS INTELIGENTES - MAPAS DE KOHONEN - SOM - SISTEMAS DE GESTIÓN DE CONTENIDOS - CMS

Abstract

This work introduces an intelligent system based on Kohonen's Maps (SOM), which can help to select - from among a predetermined group of CMSes (Content Management Systems) - the one that best fulfills a series of given requirements.

Its development was motivated by the difficulty of applying conventional selection methods, since the CMSes available on the market are too numerous and diverse.

On the generated maps, the CMSes that closest match the required features are easily identified and, from there on, conventional evaluation methods can be used.

KEYWORDS: INTELLIGENT SYSTEMS - KOHONEN'S MAPS - SOM - CONTENT MANAGEMENT SYSTEMS - CMS

Introducción

Antes de la invención de la World Wide Web, el término información solía utilizarse para incluir tanto los datos estructurados procesados mediante las aplicaciones de gestión de datos (*data management applications*), como los textos no estructurados de las aplicaciones de gestión de documentos (*document management applications*). Sin embargo, debido a la naturaleza multimedial de la Web "se hizo necesario reemplazar el término *información* por otro que abarcara, además, lo que tienen en común el audio, el video en tiempo real, el código ejecutable, la información transaccional, etc. y el término *contenido* parece servir razonablemente bien para ello" (Gilbane, 2000).

Ejemplos de *contenido* son las informaciones de la empresa, las descripciones de los productos, los catálogos, los manuales de operación, etc. que constituyen un componente fijo del lanzamiento y la ejecución de los negocios (Bechtolsheim y Oberbauer, 2001).

En cuanto a la *gestión* de los contenidos, la mayoría de las definiciones simplemente enumeran los pasos recorridos durante su procesamiento:

Existen muchas cosas que se pueden hacer con el contenido y que podrían ser, y de hecho lo son, consideradas "gestión": creación (authoring), adquisición, publicación, integración, generación dinámica de páginas, ensamblado (assembling), control de versiones (versioning), transformación, distribución, enlazado (linking), categorización, compartimiento (sharing), búsqueda (searching), sindicación (syndicating), almacenamiento temporal (caching), archivamiento, etc. (Gilbane, 2000)

Los sistemas específicos utilizados para llevar a cabo la gestión de contenidos se denominan sistemas de gestión de contenidos o CMS (*Content Management Systems*).

Tipos de sistemas de gestión de contenidos

En general, los diferentes tipos de contenido pueden ser administrados siguiendo dos enfoques opuestos: mediante sistemas que son vendidos por separado, y cuya interoperabilidad no es fácil de lograr, o mediante los llamados

enterprise content management systems (ECMS), que combinan un CMS central con otras herramientas administrativas dirigidas a todo el espectro de contenidos que existen en la organización (Robertson, 2003b).

Es oportuno destacar aquí que "existe una gran confusión en el uso del término *sistemas de gestión de contenidos empresariales* o ECMS, debido a que los proveedores de sistemas de gestión de imágenes (*imaging*), flujo de trabajo (*workflow*) o documentos, sistemas de gestión de cambios, sistemas de gestión del conocimiento (*knowledge management*), registros o portales, y sistemas para publicación en la Web, todos ellos utilizan la etiqueta ECMS, y lo hacen porque la ECM (gestión de contenidos empresariales) está de moda" (Howard, 2003).

Generalmente, la denominación *sistema de gestión de contenidos* o CMS se aplica a los sistemas cuyo fin principal es la publicación de contenidos en la Web, ya que "gestionar contenidos para la Web es el uso más común de los CMS. Sin embargo, en ciertos ámbitos, a estos sistemas se los llama *sistemas de gestión web* (web management systems o WMS)" (Robertson, 2003b).

Beneficios que ofrecen los sistemas de gestión de contenidos

Adoptar un CMS para gestionar los contenidos del sitio web de una organización trae muchos beneficios, entre los cuales pueden mencionarse:

- Reducción de costos: Las empresas han notado que implementando un sistema de gestión de contenidos, pueden asignar las tareas de publicación a los expertos en contenido (usuarios de negocio no técnicos), liberando al personal técnico para construir nuevas aplicaciones (Microsoft Corporation, 2004).
- Mejora cualitativa del sitio web: La responsabilidad del contenido en manos del usuario de negocio asegura que el contenido esté actualizado, mientras que el control centralizado del diseño asegura que el mensaje sea consistente con los valores y el *branding* de la compañía, y hace que se transmita una imagen profesional (*idem*).
- Automatización del flujo de trabajo: Dado

que se puede definir por anticipado el momento en que se actualizará automáticamente el sitio web, los cambios se pueden realizar tan pronto como se los necesite, de día o de noche (Robertson, 2003a)

- Trabajo colaborativo: Varias personas pueden actualizar el sitio web, en vez de limitar esta tarea a una sola persona (*el webmaster*). El CMS controla quién hace qué, y evita potenciales confusiones (*idem*).

- Reutilización del contenido: Una página web (o incluso un único párrafo) puede aparecer en diferentes contextos, y el CMS gestiona automáticamente su publicación en las diferentes plataformas (por ejemplo: intranet e Internet) a partir de una única fuente de contenido (lo que se conoce como *single-sourcing*) (Robertson, 2002)

- Interoperabilidad: Al estar basados en estándares de la industria, los CMS se integran fácilmente a los sistemas de negocio existentes (*idem*).

- Escalabilidad: Es cada vez más difícil encontrar personal que comprenda la tecnología y las particularidades de determinado sitio web. Mediante la separación de contenido, estructura y diseño, los CMS permiten gestionar eficientemente sitios de gran tamaño (Suh, 2002).

- Control de versiones: Un CMS permite controlar quién cambió qué y cuándo lo hizo (Robertson, 2003b)

Además de los beneficios mencionados, “el mayor beneficio que un CMS puede proporcionar es soportar las metas y estrategias del negocio. Por ejemplo, el CMS puede ayudar a mejorar las ventas, incrementar la satisfacción de los usuarios, o asistir en la comunicación con el público” (*idem*).

Proyecto para la adopción de un sistema de gestión de contenidos

La adopción de un CMS requiere de un *proyecto* que tenga como uno de sus objetivos principales la adquisición del sistema.

Un *proyecto* se puede definir como “un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. Los pro-

yectos son realizados por personas, están restringidos por la limitación de los recursos, y son planificados, ejecutados y controlados” (PMI, 2004).

El Project Management Institute define la dirección de proyectos como: *la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del proyecto. La dirección de proyectos se logra mediante la aplicación e integración de los procesos de inicio, planificación, ejecución, seguimiento y control, y cierre* (*idem*).

Los siguientes **procesos de dirección** constituyen un área de conocimiento llamada gestión de las adquisiciones del proyecto, y se requieren para comprar o adquirir productos, servicios o resultados, así como para contratar procesos de dirección (*idem*):

- **Planificar las Compras y Adquisiciones:** determinar qué comprar o adquirir, y cuándo y cómo hacerlo.

- **Planificar la Contratación:** documentar los requisitos de los productos, servicios y resultados, e identificar a los posibles proveedores.

- **Solicitar Respuestas de Proveedores:** obtener información, presupuestos, licitaciones, ofertas o propuestas, según corresponda.

- **Selección de Proveedores:** revisar ofertas, elegir entre posibles proveedores, y negociar un contrato por escrito con cada proveedor.

- **Gestión del Contrato:** gestionar el contrato y la relación entre el comprador y el proveedor, revisar y documentar cuál es o fue el rendimiento de un proveedor a fin de establecer las acciones correctivas necesarias y proporcionar una base para relaciones futuras con el proveedor, gestionar cambios relacionados con el contrato y, cuando corresponda, gestionar la relación contractual con el comprador externo.

- **Cierre del Contrato:** completar y aprobar cada contrato, incluido cualquier tema abierto, y cerrar cada contrato aplicable al proyecto.

Suh (2002) destaca que “el proceso para analizar y comprar un CMS dependerá de la escala

de las necesidades de la organización, así como el rango de productos que se evaluarán dependerá del presupuesto asignado para la gestión de contenidos (...) Para una adquisición grande, los dueños del proceso serán las gerencias de IT o de marketing, (...) mientras que para requerimientos más simples, el dueño del proceso será el profesional web”.

Sin embargo, debido a que los procesos propuestos por el PMI son genéricos y aplicables a la adquisición de cualquier producto o servicio, la mayoría de la bibliografía sobre *gestión de contenidos* propone gestionar la adquisición de un CMS a través de esos procesos u otros similares.

Bob Boiko (2001) coincide con ello al afirmar que “el proceso de selección de un proveedor de CMS no es tan diferente de cualquier otro proceso de selección de proveedores, como para que requiera de métodos propios. La mayoría de lo que propongo es sentido común para quien tiene mucha experiencia con la selección de productos”. Los ocho pasos que sugiere seguir son los siguientes:

1. Crear una breve descripción con los puntos principales del proyecto, que pueda incluirse en la correspondencia inicial con los proveedores y que sirva para orientar al comité de selección.
2. Sondear el mercado buscando los CMS que corresponden al perfil deseado. Debería obtenerse una lista con más de 10 y menos de 20 productos.
3. Realizar el primer corte de la lista de candidatos, seleccionando aquellos que merezcan consideración. No es necesario realizar un análisis demasiado exhaustivo. Utilizar una lista de entre 10 y 20 criterios con reglas de puntuación simples (rangos de puntaje cortos). La lista de candidatos luego del primer corte debería tener entre 5 y 10 CMS.
4. Establecer una lista completa de criterios de selección y un mecanismo de puntuación, darles forma de *solicitud de propuesta y enviárselas* a los proveedores que pasaron el primer corte.
5. Seleccionar un número pequeño de fina-

listas (entre 2 y 5), calificando para ello las respuestas a las *solicitudes de propuesta*.

6. Llevar a cabo encuentros con los finalistas y verificar sus referencias.
7. Pedirles a los finalistas que realicen una presentación o demostración de sus CMS.
8. Tomar la decisión final combinando los puntajes de las solicitudes de propuesta con los de las referencias y la presentación.

Para evitar que el ganador del proceso de selección sea descartado “por cualquier razón”, es necesario que, antes de comenzar, “todos estén de acuerdo con el proceso y con que gane el que resulte ganador” (idem). Por ello, los mecanismos de puntuación deben definirse por anticipado, a fin de realizar una evaluación justa e imparcial.

Bob Doyle (2004a) menciona 15 pasos para seleccionar un CMS, aunque aclara que “algunos son demasiado caros para ciertos presupuestos y otros requieren más tiempo de estudio del que se dispone”:

1. Organizar el contenido.
2. Buscar información en la Web.
3. Leer libros y artículos sobre el tema.
4. Contratar consultores neutrales.
5. Detectar las compañías con publicidad en revistas especializadas.
6. Leer reportes de los analistas de la industria.
7. Visitar demostraciones comerciales.
8. Contactar a los proveedores.
9. Contratar consultores específicos de los productos.
10. Identificar requerimientos y enviar solicitudes de propuesta.
11. Utilizar demos para conocer los sistemas.
12. Realizar el primer corte para obtener una lista de 2 a 5 productos.
13. Pedirles a los finalistas un prototipo realizado con sus CMS.
14. Pedirles a los finalistas una cronograma estimativo para migrar al CMS.
15. Seleccionar el CMS que será adquirido.

A pesar del gran número de pasos, esta metodología no difiere sustancialmente de la anterior, ya que los ocho pasos del 2 al 9 podrían englobarse bajo el título “Sondear el mercado en busca de CMS”.

Métodos convencionales para la evaluación de un conjunto de CMS.

Al responder a las solicitudes de propuesta que contienen los requerimientos de gestión de contenidos de la organización, los proveedores ofrecen información detallada sobre cómo sus CMS cumplirán con cada uno de los requerimientos.

En base a esta información, es posible evaluar los CMS y compararlos entre sí, a fin de seleccionar el sistema ganador.

En la tabla 1 se mencionan algunos de los métodos utilizados más frecuentemente para realizar la evaluación y/o selección de software, y que, por lo tanto, se aplican también en el caso particular de los CMS.

Suma y ponderación numéricas (NWS o <i>Numerical Weight and Sum</i>)	Mide la calidad de un producto de software como la suma ponderada de sus atributos (Anderson, 1990).
Suma y ponderación cualitativas (QWS o <i>Qualitative Weight and Sum</i>)	Utiliza símbolos como pesos. Para armar el ranking, se cuenta cuántos símbolos de cada tipo obtuvo cada producto (Scriven, 1991).
Maximax (MM)	Se ordenan los productos según su máximo puntaje (no importa en cuál atributo se obtuvo) (Anderson, 1990).
Eliminación por aspectos (EBA o <i>Elimination by aspects</i>)	Se eliminan sucesivamente los productos que no alcanzan el puntaje mínimo para cada atributo (idem).
Ordenamiento lexicográfico (LO o <i>Lexicographic Ordering</i>)	Los productos se ordenan en base al puntaje que cada uno obtuvo en el atributo más importante, luego en el siguiente, etc. (idem)
Agregación lógica de preferencias (LSP o <i>Logic Scoring of Preferences</i>)	Es un modelo cuantitativo de decisión para la evaluación, comparación y selección de sistemas de hardware y software complejos, basado en una lógica de preferencias continua (Dujmovic, 1996).

Tabla 1. Métodos para realizar la evaluación y/o selección de software

Selección de un CMS a través de la Web

Varias organizaciones han colocado sistemas

en la Web con el fin específico de facilitar la selección de un CMS. En la tabla 2 se describen brevemente cuatro de estos sistemas¹.

CMS-Search http://www.cmsreview.com/CMSSearch.html	
Fabricante	La asociación <i>OSCOM</i> , el sitio web <i>CMS Review</i> , la consultora <i>Hartman Communicatie BV</i> , la organización <i>CM Pros</i> y el laboratorio de evaluación de CMS de la <i>Universidad de Washington iSchool</i> .
Tamaño	Dispone de una base con las características de 73 CMS.
Puntos fuertes	Utiliza CMSML (CMS Markup Language), un lenguaje diseñado para representar las características y funciones principales de los CMS.
Puntos débiles	Muchas de las características deseadas en el CMS a encontrar deben ingresarse como texto y no hay indicaciones de cuáles valores son posibles. No proporciona una explicación contextual de los términos que emplea. Su implementación está incompleta.

Tabla 2. Sistemas disponibles en la Web para la selección de un CMS

¹ Los sistemas descritos fueron analizados en enero de 2008. Actualmente, sus características pueden diferir de las expuestas aquí.

Produktfinder http://www.contentmanager.de/itguide/produktfinder.html	
Fabricante	La agencia alemana de publicidad FEIG & Partner.
Tamaño	Dispone de una base con las características de 1419 CMS.
Puntos fuertes	Los criterios de búsqueda se eligen mediante casillas de verificación (<i>checkboxes</i>) y botones de exclusión mutua (<i>radio buttons</i>).
Puntos débiles	Sólo ofrece explicaciones contextuales en la página de inicio. A pesar de clasificar los CMS en varias categorías, los resultados de las búsquedas no parecen considerarlas (pueden obtenerse listados con más CMS que los existentes en la categoría consultada).
Content Management Overview http://tools.hartman-communicatie.nl/overzicht.html	
Fabricante	Hartman Communicatie BV, una consultora neerlandesa independiente.
Tamaño	Dispone de una base con las características de 174 CMS.
Puntos fuertes	Recién cuando son seleccionados los criterios, se despliegan los controles - casillas de verificación (<i>checkboxes</i>), botones de exclusión mutua (<i>radio buttons</i>) y menús desplegables - que permiten elegir los subcriterios.
Puntos débiles	El diseño utilizando menús desplegables impide la búsqueda de un CMS con dos características que estén presentes en un mismo menú. Las explicaciones contextuales sólo aparecen en las vistas de los productos (recién pueden consultarse <i>después</i> de haber efectuado una búsqueda).
CMS-Matrix http://www.cmsmatrix.org/matrix/cms-matrix?func=search	
Fabricante	La empresa americana Plain Black Corporation, creadora del CMS <i>WebGUI</i> .
Tamaño	Dispone de una base con las características de 873 CMS.
Puntos fuertes	Ofrece explicaciones contextuales emergentes para cada característica. Muchas características se eligen con casillas de verificación (<i>checkboxes</i>).
Puntos débiles	Algunas de las características deseadas en el CMS a encontrar deben ingresarse como texto y no hay indicaciones de cuáles valores son posibles. No es posible acceder simultáneamente a los datos de más de 10 CMS.

Tabla 2. (continuación) Sistemas disponibles en la Web para la selección de un CMS

El problema de la identificación de los posibles proveedores

En cualquier *proyecto* dirigido a la adopción de un CMS existen procesos de dirección pertenecientes al área de conocimiento que el Project Management Institute denomina *gestión de las adquisiciones del proyecto*, y que se requieren para llevar a cabo la compra o adquisición del CMS. Uno de estos procesos es la *Planificación de la Contratación*, entre cuyos subprocesos se encuentra la identificación de los posibles proveedores para solicitarles información detallada sobre sus productos (a través del proceso denominado *Solicitar Respuestas de Proveedores*). Esta información servirá de base, posteriormente, para realizar la selección del CMS que mejor cumple con los requerimientos de la organización (mediante

el proceso denominado *Selección de Proveedores*).

En la práctica, la identificación de los posibles proveedores se reduce a obtener una lista que contenga los nombres de 10 a 20 posibles proveedores como recomienda Boiko (2001) o de 2 a 5 posibles proveedores como sugiere Doyle (2004a).

Los responsables de llevar a cabo la identificación de los posibles proveedores encuentran dos serias dificultades:

- Deben llegar a un número relativamente pequeño de sistemas partiendo de un conjunto inicial formado por los CMS existentes en el mercado. El número de CMS que aparecen en los principales directorios de la Web

varía enormemente, pero “después de copiar y pegar los listados, y de filtrarlos para eliminar la redundancia, es posible obtener los nombres de cerca de 1800 CMS” (Doyle, 2005).

- La cantidad de propiedades que caracterizan a un CMS es problemática, ya que su número también es elevado. Por ejemplo, el *Content Management Requirements Toolkit* es un listado que “contiene 133 requerimientos totalmente desarrollados, listos para copiar y pegar en su solicitud de propuestas” (STD, 2004).

Una vez que se dispone de un conjunto reducido de proveedores de CMS, la finalización del proceso de selección del sistema que mejor se adapta a las necesidades de la organización se puede realizar aplicando métodos convencionales de evaluación.

Considerando las metas y los factores mencionados, el problema que se propone resolver puede ser enunciado de la siguiente manera:

Dados:

A: un conjunto de n CMS, cada uno representado por sus c características presentes o ausentes (Fig. 1)

R: una lista de requerimientos representada por (como máximo) c características (Fig. 2)

Se desea ordenar los n CMS pertenecientes al conjunto A de forma tal que sea posible identificar fácilmente aquellos que, por tener una mayor afinidad con los requerimientos de R, serían los más apropiados para participar de un proceso de selección posterior.

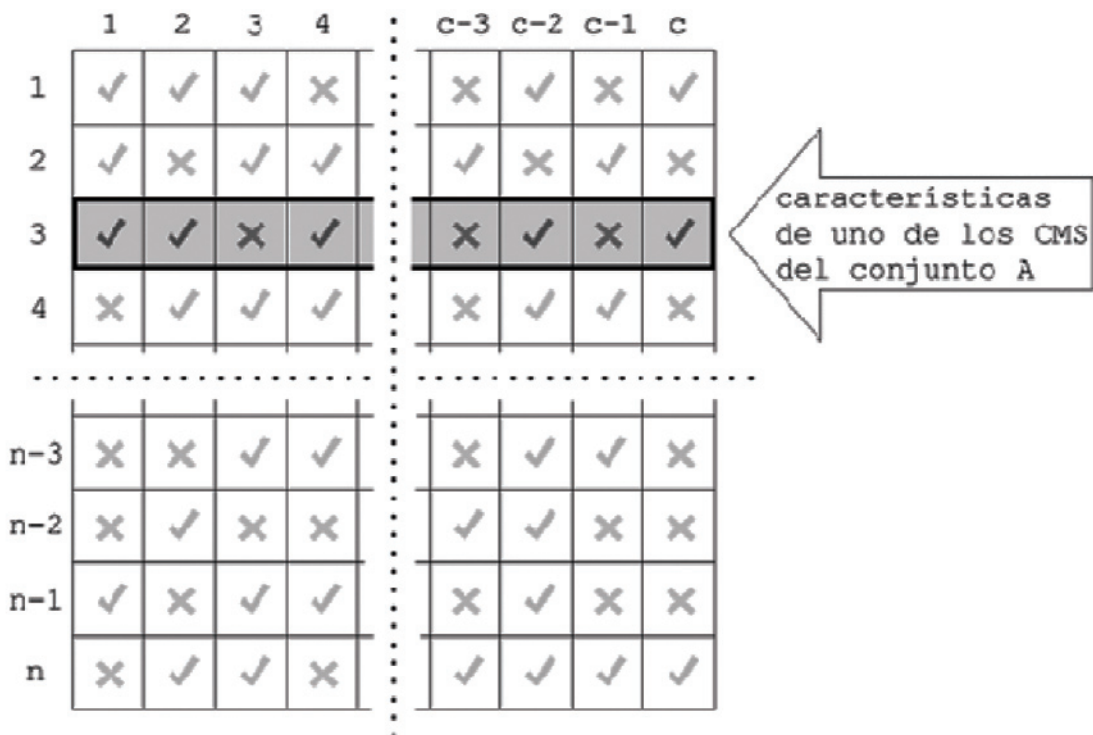


Fig. 1. A: un conjunto de n CMS, cada uno representado por sus c características



Fig. 2. R: una lista de requerimientos (c características como máximo)

Debido a que en el caso de los CMS los atributos a ponderar son tan numerosos, los métodos convencionales de evaluación no funcionarían correctamente. Tampoco los sistemas que varias organizaciones han colocado en la Web con el fin específico de facilitar la selección de un CMS permiten resolver satisfactoriamente el problema. Algunas de las dificultades que presentan estos sistemas son:

- **Fallas en el diseño de la interfaz:** si los requerimientos deben ingresarse como texto, es difícil saber qué ingresar.

- **Ingreso y edición de datos sin moderador:** Si los visitantes de la página web de un sistema pueden modificar los datos de los CMS, la información pierde confiabilidad. Esto explica por qué los sistemas no moderados contienen los datos de más de 800 CMS, mientras que los sistemas con moderador sólo contienen los datos de menos de 200 CMS.

- **Especificación estricta de requerimientos:** Los controles que sólo permiten elegir entre dos estados (sí/no), provocan que sean descartados los CMS que no cumplen todos los requerimientos, con lo cual, a veces, la búsqueda no arroja resultados. En tales casos, podría ser mejor para los usuarios obtener los resultados que más se acerquen a sus requerimientos, aunque no los cumplan completamente.

- **Imposibilidad de comparar los resultados:** Si no es posible comparar las características de los CMS que cumplen con los requerimientos de los usuarios, se dificulta la posterior selección.

Por lo tanto, la solución buscada sólo se puede obtener mediante un enfoque diferente de los convencionales. A continuación, se presenta un *Sistema Inteligente* que resuelve satisfactoriamente el problema planteado.

Parte Experimental

Para resolver el problema planteado se desarrolló *CMS-SOM*, un *Sistema Inteligente* cuyo núcleo está constituido por una red neuronal. Las redes neuronales son un modelo computacional basado en unas "unidades de procesamiento sorprendentemente simples" (Dayhoff, 1990) denominadas neuronas artificiales, in-

terconectadas y funcionando en paralelo.

Las **características más relevantes** de *CMS-SOM* son las siguientes:

- El sistema emplea mapas autoorganizados de Kohonen (SOM²) y, por lo tanto, entra dentro de la categoría de los *Sistemas Inteligentes*.
- El sistema es de fácil uso: el usuario ingresa sus requerimientos y recibe un mapa donde se encuentran distribuidos, en un panel de celdas hexagonales, los nombres de un grupo predefinido de 160 CMS y una leyenda representando los requerimientos ingresados. Cuanto más cerca de la leyenda aparezca un CMS, mayor será la aptitud de éste.
- El sistema puede instalarse para ser accesible a través de la Web mediante un navegador (*browser*).
- El sistema mantiene sus datos en un fichero en XML (*Extensible Markup Language*), lo que facilita su procesamiento (lectura, edición, etc.)
- El sistema está formado por *Java servlets*, que generan dinámicamente las páginas vistas en el navegador, a partir de los requerimientos ingresados por el usuario, de los datos de los CMS leídos desde el fichero en XML y de los cálculos que se realizan.
- El sistema emplea hojas de estilo CSS (*Cascade Style Sheets*) para definir el formato de los distintos elementos de las páginas web.
- El sistema cumple con los estándares Web del *Consortio W3C*.
- El sistema sólo permite que sea el administrador quien modifique los datos de los CMS, lo que ayuda a garantizar la veracidad de los mismos.
- El sistema ofrece una gran flexibilidad para ingresar los requerimientos, que pueden estar formados por hasta 400 características (agrupadas en 66 categorías, y éstas a su vez en 10 rubros) que definen a los CMS.

Constituyen **características originales** de *CMS-SOM*:

- El sistema ofrece una tabla completa con los datos de los CMS, para su comparación. Otros sistemas no muestra los datos con que trabajan.
- El sistema permite especificar requerimientos de 5 tipos: "obligatoriamente NO", "preferentemente NO", "indistinto", "preferentemente SÍ" y "obligatoriamente SÍ". Otros sistemas sólo permiten es-

²SOM es el acrónimo de *Self-Organizing Map*

tablecer un único tipo de requerimientos (aquellos que obligatoriamente deben cumplirse).

- El sistema utiliza solamente botones de exclusión mutua (*radio buttons*), por lo que nunca es necesario ingresar textos. Otros sistemas utilizan métodos menos eficaces para el ingreso de los requerimientos.

- El sistema muestra explicaciones contextuales de todos los requerimientos efectuables, para que el usuario siempre esté informado del significado de las distintas opciones. Otros sistemas requieren profundos conocimientos sobre gestión de contenidos para ser operados.

Los mapas autoorganizados de Kohonen (SOM)

Los mapas autoorganizados, desarrollados por Teuvo Kohonen durante la década de 1980 en la Universidad Tecnológica de Helsinki, se destacan porque "de entre todos los modelos de red neuronal, probablemente sea el que mejor modela lo que ocurre realmente en el cerebro" (Welstead, 1994).

Sin embargo, hoy en día las redes neuronales artificiales solamente tienen una importancia marginal como modelo de funcionamiento del cerebro humano, ya que son "consideradas estrictamente como interesantes y útiles dispositivos de ingeniería" (Nilsson, 1998).

Una aplicación práctica de los mapas autoorganizados de Kohonen es "encontrar categorías (*clusters*) en la información de entrada y que luego un vector de datos desconocido sea identificado con una de las categorías" (Kohonen, 1996). Eso es precisamente lo que hace el sistema *CMS-SOM*.

Arquitectura de SOM

En comparación con otros tipos de redes neuronales, la arquitectura de SOM es extremadamente sencilla. En este modelo, la red neuronal está formada solamente por dos capas (Fig. 3):

- Capa de entrada: Es la capa en la que se le presenta a la red la información de entrada contenida en un vector $x = (\xi_1, \xi_2, \dots, \xi_n) \in \mathfrak{R}^n$.

- Capa de salida: Es la capa donde la red neuronal muestra su respuesta ante la presentación de cierta entrada. Las neuronas de esta capa forman un arreglo de forma "rectangular, hexagonal, o incluso irregular, siendo el formato hexagonal el visualmente más efectivo" (Kohonen, 2001). Cada neurona i de esta capa tiene asociado un vector modelo $m_i = (\mu_{i1}, \mu_{i2}, \dots, \mu_{in}) \in \mathfrak{R}^n$ (gráficamente, el valor μ_{ij} representaría una conexión entre una neurona i de salida y una neurona j de entrada).

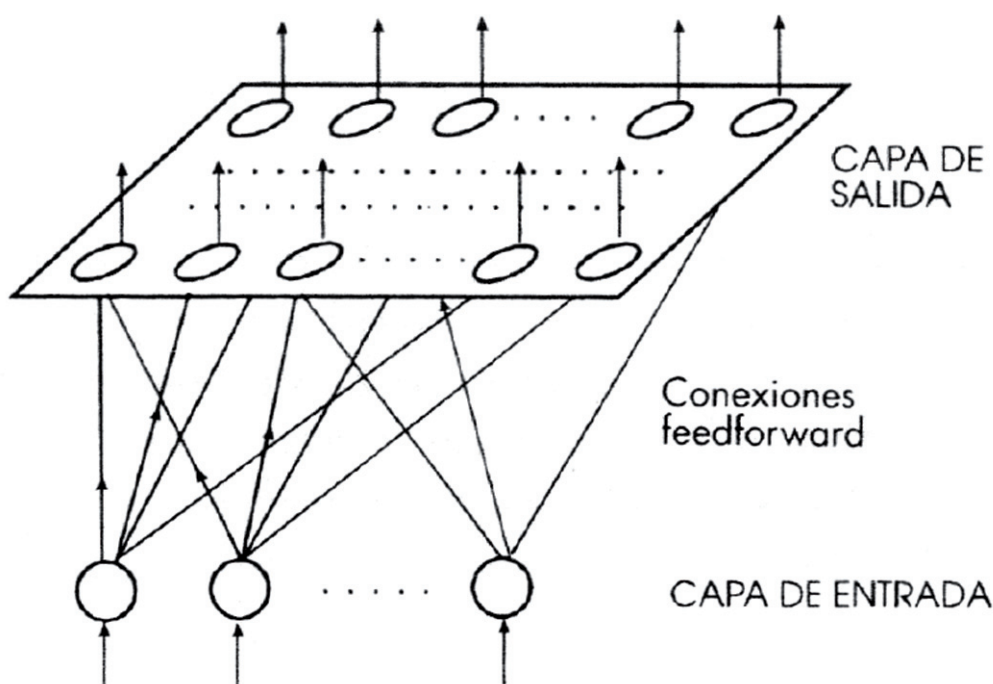


Fig. 3. Arquitectura de la red neuronal SOM (Hilera y Martínez, 1995)

Funcionamiento de SOM

El objetivo de la red neuronal SOM es que, al recibir un vector con la información de entrada, éste sea procesado y, como respuesta, se obtenga cuál es la neurona de salida que corresponde al vector ingresado. Según Kohonen (2001), "la *magnitud* exacta de la respuesta no precisa ser calculada: la entrada simplemente es mapeada en esa *ubicación*".

El procesamiento que ocurre con el vector de entrada x para determinar cuál es c , la neurona con la que se lo debe mapear como respuesta, "es una tarea trivial si la red neuronal es simulada mediante un programa de computadora" (idem). Lo que se realiza es la comparación del vector de entrada x con los vectores modelo m_i de todas neuronas de salida i , mediante alguna métrica, y c es la neurona i que mejor resulta en la comparación.

Usualmente, la métrica que se utiliza para comparar el vector de entrada x con los vectores modelo m_i es el valor mínimo de las distancias euclídeas $\|x - m_i\|$, por lo que la neurona c es la que satisface la igualdad:

$$\|x - m_c\| = \min_i \|x - m_i\|$$

La fórmula de la distancia euclídea es la siguiente, aunque también "suele utilizarse la expresión eliminando la raíz cuadrada" (Hilera y Martínez, 1995):

$$\|x - m_i\| = \sqrt{\sum_j (x_j - m_{ij})^2}$$

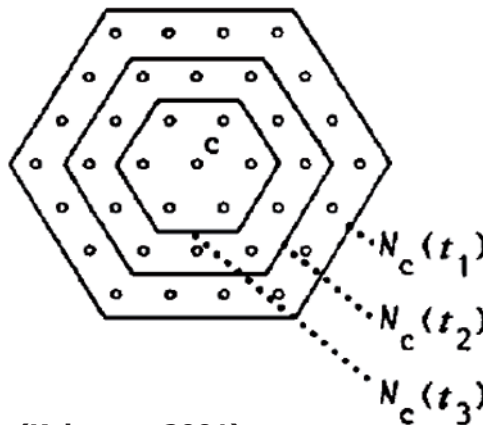


Fig. 4. Vecindad topológica (Kohonen, 2001)

³ Los pasos se deben repetir "un número razonablemente grande de veces (...) Para un aprendizaje rápido (...) 10 000 iteraciones o menos pueden ser suficientes" (Kohonen, 2001)

⁴ "Por aprox. 1000 iteraciones, U debe tener valores cercanos a la unidad (...) luego del periodo inicial de ordenamiento, debería mantenerse en valores del orden de 0.02 o menores" (idem)

Aprendizaje de SOM

En las redes SOM, la *etapa de funcionamiento*, que es cuando se le presenta a la red neuronal un vector de entrada x para que lo mapee a una neurona de salida c , ocurre luego de haber finalizado la etapa de aprendizaje, que es cuando se ajustan los valores de los vectores modelo m_i . Por ello, "el aprendizaje en el modelo SOM es de tipo OFF LINE" (Hilera y Martínez, 1995).

Otra característica de este modelo es que utiliza un aprendizaje no supervisado. A la red neuronal SOM "sólo se le proveen valores de entrada, y se le requiere que les dé sentido según su propio criterio" (Welstead, 1994).

El algoritmo de aprendizaje de SOM es "computacionalmente muy liviano" (Kohonen, 2001), y consta de los siguientes pasos (Hilera y Martínez, 1995):

1. Se inicializan los vectores modelo m_i con valores aleatorios (p. ej. entre -1 y 1), y se fijan los valores de los tres parámetros de aprendizaje:

- **Número de iteraciones para los pasos 2-4³**

- **Factor de aprendizaje D^4**

- **Radio de la zona de vecindad.**

La zona de vecindad N_c abarca la neurona c y las neuronas de salida ubicadas alrededor de ella. El radio debe ir disminuyendo al avanzar el aprendizaje (Fig. 4).

2. A continuación se presenta a la red una información de entrada (la que debe aprender) en forma de vector $x = (\xi_1, \xi_2, \dots, \xi_n) \in \mathfrak{R}^n$.

3. Puesto que se trata de un aprendizaje competitivo, se determina c , la neurona vencedora de la capa de salida.

4. Una vez localizada c , la neurona vencedora, se actualizan los vectores modelo m_i aplicando la siguiente fórmula, donde $t = 0, 1, 2, \dots$ es un entero, el número de la iteración:

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot [x(t) - m_i(t)] \text{ para } i \in N_c(t)$$

Los ajustes realizados en el último paso "hacen que la neurona ganadora y sus vecinas se vuelvan más parecidas al vector de entrada x . De esta forma, la neurona ganadora tendrá más probabilidades de ganar la próxima vez que se presente el mismo vector de entrada u otro similar" (Dayhoff, 1990)

Desarrollo del sistema CMS-SOM

Para desarrollar CMS-SOM, debió elegirse el paradigma de la ingeniería del software que se seguiría. Este puede definirse como "una estrategia de desarrollo que abarca procesos, métodos y herramientas, y se elige de acuerdo a la naturaleza del proyecto y la aplicación, los mé-

todos y herramientas a utilizar, y los controles y entregables requeridos" (Pressman, 2001).

Entonces, por ser un término medio entre comenzar a programar directamente sin ninguna planificación previa⁵ y los complejos procesos de análisis y diseño que incluyen cientos de páginas de documentación, para el desarrollo de CMS-SOM se eligió seguir la metodología de desarrollo orientado a objetos para proyectos pequeños propuesta por Carlos Fontela (2003).

Una vez programado el sistema CMS-SOM, fue necesario proporcionarle (cargándolos en un fichero en XML) una lista de características descriptivas de los CMS y los datos de un grupo predeterminado de éstos.

Una nueva lista de características descriptivas de CMS

Para CMS-SOM se diseñó un listado de 10 rubros (Tabla 3) que fueron luego divididos en 66 categorías, en las cuales, finalmente, fueron agrupadas las 400 características que definen a los CMS. En base a esta información se definió la estructura del fichero *items.xml*, que es el fichero leído por CMS-SOM.

Rubro	Denominación
1	Fabricante
2	Generalidades del producto
3	Licencia
4	Documentación
5	Soporte
6	Tecnologías
7	Estándares
8	Ciclo de Vida del Contenido: CREACIÓN
9	Ciclo de Vida del Contenido: GESTIÓN
10	Ciclo de Vida del Contenido: PUBLICACIÓN

Tabla 3. Rubros en que CMS-SOM clasifica las características de los CMS

⁵ Es lo que propone el modelo code-and-fix (codificar y corregir), que "es un modelo poco útil pero bastante común" (McConnell, 1996,140)

Los tres últimos rubros corresponden a los pasos recorridos durante el procesamiento del contenido, los cuales constituyen el "ciclo de vida del contenido" (Fig. 5). Inicialmente, tiene lugar la *creación* del contenido. Luego, éste debe ser sometido al control de los responsables hasta su aprobación (*gestión del contenido*). A continuación, es posible realizar su *publicación*. La posibilidad de cambiar la apariencia del contenido publicado (por ejemplo, la escala utilizada para mostrar una imagen), puede considerarse como parte del proceso de publicación o como un paso independiente: la *presentación*.

Elección de los CMS abarcados por CMS-SOM y obtención de sus datos

Para elegir cuáles serían los CMS considerados en CMS-SOM, se siguió el siguiente procedimiento:

1. Se realizó la unión de los conjuntos de CMS abarcados en *CMS-Search*, *Produktfinder*, *CM-Overview* y *CMS-Matrix*.
2. Se eliminaron los CMS que sólo aparecían en uno de los conjuntos.
3. Se revisó la lista de los CMS resultantes,

actualizando los nombres de aquellos que cambiaron y eliminando los que ya no están más en el mercado.

4. Finalmente, se agregaron dos CMS más⁶, obteniéndose la lista definitiva de 160 CMS abarcados por *CMS-SOM*.

La obtención de los datos de esos 160 CMS se realizó siguiendo dos metodologías diferentes.

En primer lugar, se intentó contactar a los fabricantes para solicitarles que colaboraran con este trabajo, a través del aporte de los datos referidos a sus CMS. Se les envió un mensaje de correo electrónico con un código distinto a cada uno de ellos, invitándolos a llenar un formulario *online*.

Cuando no fue posible conseguir los datos de un CMS de esa manera, se emplearon dos técnicas distintas para buscar los datos de los CMS en la Web:

- Extracción automática de los datos disponibles en *CM-Overview* y *CMSMatrix*.
- Búsqueda manual de los datos disponibles en *CMS-Search*, *Produktfindery* las páginas web de los propios CMS.

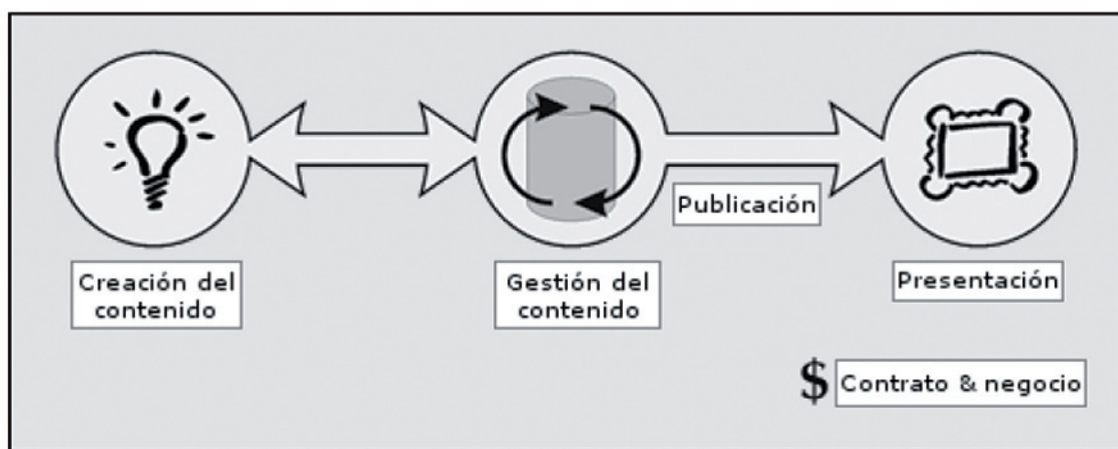


Fig. 5. El ciclo de vida del contenido (Robertson, 2003b)

⁶ El CMS *360 Web Manager 3.0*, a pesar de ser mencionado solamente en *CMS-Matrix*, fue agregado a la lista porque se consideró interesante incluir un producto argentino en CMS-SOM (también es argentino el autor del *CMS TikiWiki*, actualmente desarrollado por voluntarios de todo el mundo), y el sistema *c-BIZZ*, mencionado solamente en *Produktfinder*, se agregó por ser el CMS con que habitualmente trabajaba el autor.

Resultados y Discusión

Los resultados que se presentan aquí sirven para probar que los mapas autoorganizados que se obtienen mediante *CMS-SOM* son correctos.

Vale aclarar que, cuando el usuario ingresa alguna característica como *Indistinto*, se construyen dos mapas: uno abarcando todas las características posibles y otro considerando solamente las características requeridas.

Por ello, hubo que verificar que ambos tipos de mapas fueran correctos.

Mapas autoorganizados abarcando todas las características posibles

La validez de los mapas autoorganizados construidos al considerar las 400 características posibles se determinó analizando algunos de ellos y verificando si la distribución de los CMS en

cada mapa se puede reconocer como correcta.

Para ello, se accedió cuatro veces al sistema *CMS-SOM* solicitando todas las características como *Indistinto*. En los mapas autoorganizados obtenidos se marcó luego la ubicación de los siguientes tres grupos de CMS:

- **Grupo 1:** PHP-Nuke 8.0, PostNuke 0.764 y Xaraya 1.1.2
- **Grupo 2:** Mambo 4.6.1 y Joomla! 1.0.11
- **Grupo 3:** Drupal 4.7 y Typo3 - Version 4.0

Se concluyó que la distribución de los CMS en los mapas autoorganizados analizados puede considerarse correcta, ya que los CMS de cada uno de los tres grupos permanecieron siempre juntos en las cuatro oportunidades, algunas veces en la misma celda, y otras veces en celdas contiguas o cercanas. Esto se debe a que los CMS de cada grupo poseen características similares (Tabla 4).

PHP-Nuke 8.0, PostNuke 0.764 y Xaraya 1.1.2
El hecho de que PHP-Nuke 8.0 y PostNuke 0.764 posean similares características y que, por ello, hayan sido ubicados siempre juntos en los mapas por <i>CMS-SOM</i> , se debe probablemente a que PostNuke es una bifurcación (<i>fork</i>) de PHP-Nuke originada en el código fuente de éste, por lo cual, inevitablemente, mantiene muchas de sus características. Xaraya 1.1.2 y PostNuke 0.764 también son sistemas de características similares, por el mismo motivo mencionado en el caso anterior: Xaraya es una bifurcación de PostNuke.
Mambo 4.6.1 y Joomla! 1.0.11
Los CMS pertenecientes al segundo grupo identificado en los mapas autoorganizados también se mantuvieron juntos en las cuatro pruebas realizadas. La causa probable por la cual ambos CMS son similares, es que Joomla! surgió como una bifurcación de Mambo, en 2005.
Drupal 4.7 y Typo3 - Version 4.0
A pesar de ser dos desarrollos independientes, ambos aparecieron juntos en los cuatro mapas. Buscando una explicación, se realizaron dos búsquedas con <i>Google</i> en las cuales puede verse que Drupal y Typo3 realmente presentan una fuerte tendencia a aparecer juntos. Al buscar <i>Drupal Typo3</i> , se encontraron aprox. 833 000 páginas web, con títulos como "Drupal versus Typo3", "Se busca desarrollador Drupal/Typo3" o "Typo3 vs. Drupal". La búsqueda de la combinación inversa <i>Typo3 Drupal</i> arrojó un resultado de aprox. 832 000 páginas y títulos parecidos a los anteriores. Evidentemente, no fue casual que <i>CMS-SOM</i> colocara a Drupal 4.7 y a Typo3 siempre en celdas próximas. Estudiando ambos sistemas en detalle, se encuentran muchas semejanzas: los dos están disponibles con la licencia GNU/GPL, su lenguaje de programación es el PHP, soportan varias bases de datos, entre ellas MySQL, así como los servidores web Apache e IIS.

Tabla 4. Análisis de la ubicación de los CMS de cada grupo en los mapas

Mapas autoorganizados considerando sólo características requeridas

La validez de los mapas autoorganizados construidos por *CMS-SOM* considerando solamente las características que le son requeridas se determinó analizando algunos de ellos y verificando que la distribución de los CMS en cada mapa fuera compatible con los resultados obtenidos al consultar los otros cuatro sistemas disponibles en la Web para facilitar la selección de un CMS: *CMS-Search*, *Produktfinder*, *CM-Overview* y *CMS-Matrix*.

Para ello, se accedió al sistema *CMS-SOM* estableciendo como requerimientos los siguientes grupos de características:

- **Primera prueba:** Se solicitaron CMS que obligatoriamente fueran gratuitos y estuvieran basados en la tecnología LAMP (sistema operativo Linux, servidor web Apache, base de datos MySQL y lenguaje PHP).

- **Segunda prueba:** El objetivo fue seleccionar CMS con un precio obligatoriamente menor que USD 5 000, basados en la tecnología J2EE, que funcionaran con el servidor de aplicaciones IBM Websphere y que, además, preferentemente permitieran importar ficheros DOC y editar contenidos usando la técnica WYSIWYG (*what you see is what you get*).

- **Tercera prueba:** La meta fue seleccionar CMS que obligatoriamente costaran entre USD 10 000 y USD 20 000, estuvieran basados en la tecnología .NET y usaran bases de datos MS SQL Server, y que preferentemente permitieran la publicación de versiones del contenido accesibles para usuarios con discapacidades.

- **Cuarta prueba:** Se solicitaron CMS que preferentemente no costaran más de USD 500, que estuvieran basados únicamente en el lenguaje Perl y fueran obligatoriamente compatibles con los sistemas operativos Linux y Windows Server 2003, que preferentemente ofrecieran una vista previa antes de publicar los contenidos y con los cuales obligatoriamente se pudieran realizar foros y encuestas.

- **Quinta prueba:** El objetivo fue seleccionar CMS que obligatoriamente estuvieran basados en la tecnología Zope, y en los cuales la generación de los documentos a publicar fuera rea-

lizada preferentemente por un *Live Server*.

- **Sexta prueba:** Se solicitaron las características exactas del CMS Jahia.

Los resultados de las 6 pruebas fueron favorables a *CMS-SOM*, aunque con distintos grados de contundencia.

En la primera prueba, 25 CMS resultaron ubicados en la celda correspondiente a los requerimientos efectuados, por lo que no se estudió lo que ocurrió en las celdas adyacentes o cercanas. Cabe destacar que 25 CMS gratuitos basados en LAMP representan el 15,6% de los 160 CMS considerados por *CMS-SOM*, lo que representa un porcentaje significativamente mayor que el 2,4 ~ 6,3% con que estos CMS están presentes en los resultados de los otros sistemas. Esos bajos porcentajes observados no se deben a que los CMS gratuitos basados en LAMP no figuran en los demás sistemas, sino al hecho de que no son mostrados porque no coinciden con los criterios de búsqueda. Al rastrear la causa de esta discordancia en los resultados, se detectó que la falla se originó por el deficiente ingreso de requerimientos. Por ejemplo, al solicitar en *CMS-Matrix* un CMS que corra bajo Linux, no resultan seleccionados los CMS que como valor de su característica "Sistema Operativo" tienen registrado "any" (que en inglés significa "cualquiera"). Además, hubo 4 CMS que sólo fueron propuestos por *CMS-SOM*. Accediendo a las páginas web de estos productos, pudo verificarse que los cuatro son CMS gratuitos basados en LAMP, por lo que fueron correctamente sugeridos por *CMS-SOM* e incorrectamente ignorados por los otros sistemas.

En la segunda prueba, resultaron 3 CMS ubicados en la misma celda que los requerimientos efectuados (*Liferay Portal*, *Magnolia* y *MMBase*) y 6 CMS más aparecieron en tres de las celdas adyacentes (*Oracle Universal CM* en la celda de arriba a la izquierda, *EPAM CMS* y *FeedStream QDoX* en la celda de la derecha, y *CuppaWeb*, *Simplicis Marketing Dashboard* y *SR2* en la celda de abajo a la derecha). La tabla obtenida luego de generar los mapas autoorganizados permite observar que, de los CMS mencionados, 5 califican, aunque sólo *Liferay Portal* cumple exactamente con los requerimientos efectuados, porque los otros 4 CMS no satisfacen una o más preferencias (o se desconoce si

lo hacen). Comparar los resultados obtenidos mediante *CMS-SOM* en la segunda prueba con los que surgieron al consultar *CMS-Search*, *Produktfinder*, *CM-Overview* y *CMS-Matrix* no resultó ser una tarea trivial, ya que estos sistemas no permiten ingresar la totalidad de las características requeridas en esta prueba. En particular, los requerimientos que sólo son preferencias pero no son obligatorios, no deben ingresarse en estos sistemas, ya que siempre son considerados como características obligatorias. Esta vez, *CMS-SOM* produjo resultados en gran parte coincidentes con los de *CM-Overview* (CuppaWEB, Magnolia, MMBase y Simplicis Marketing Dashboard). La falta de resultados de *CMS-Search* y la coincidencia nula entre las sugerencias de *Produktfinder*, *CMS-Matrix* y *CMS-SOM* podrían explicarse por el tipo de requerimientos efectuados, pero, en realidad, vuelve a ponerse en evidencia que la verdadera causa reside en el problemático modo de ingreso de datos provisto por *CMS-Search*, *Produktfinder* y *CMS-Matrix*.

En la tercera prueba, en la celda correspondiente a los requerimientos efectuados aparecieron 10 CMS: activeWeb contentserver 5.5, Digimaker 5.2, Applaud CMS 3.5, Clay Tablet Rosetta WCMS, EPiServer 4.60, SR2 v7.0, Estrada Engine 3.5, Noxum Publishing Studio, Numotion WebManager y Web500 CMS. También en esta prueba, el elevado número de CMS obtenidos en la celda correspondiente a

los requerimientos efectuados hizo que no fuera necesario estudiar los CMS de las celdas adyacentes, y pudo considerarse satisfactorio el desempeño de *CMS-SOM* simplemente comparando la lista de los CMS que éste sugirió con los propuestos por los otros sistemas. Para cada uno de los cinco sistemas, nuevamente se calculó la relación entre el número de CMS que cumplen los requerimientos efectuados y el número de CMS considerados. Es interesante destacar que, una vez más, *CMS-Search* no arrojó ningún resultado, por lo que su relación es de 0%. Esto casi seguro se deba a algún defecto en su sistema de búsqueda, porque es muy poco probable que el resultado de las dos últimas pruebas se deba a la inexistencia de CMS con las características requeridas entre los CMS de su registro. Por la imposibilidad de expresar en *CMS-Matrix* un precio requerido de entre USD 10 000 y USD 20 000, este sistema sugirió CMS basados en .NET y que utilizan MS SQL Server, pero cuyos precios están fuera del rango solicitado (por ejemplo: Composite CMS, Kentico CMS y Microsoft CMS).

En la cuarta prueba, solamente aparecieron dos CMS (Metadot Portal Server 6.4.5 y WebGUI 7.0) en la celda de los requerimientos efectuados (Fig. 6). En las celdas adyacentes aparecen otros tres CMS (a la izquierda Liferay Portal 4.2 y arriba a la derecha Hippo CMS 6.3.6 y Silva 1.5.9).

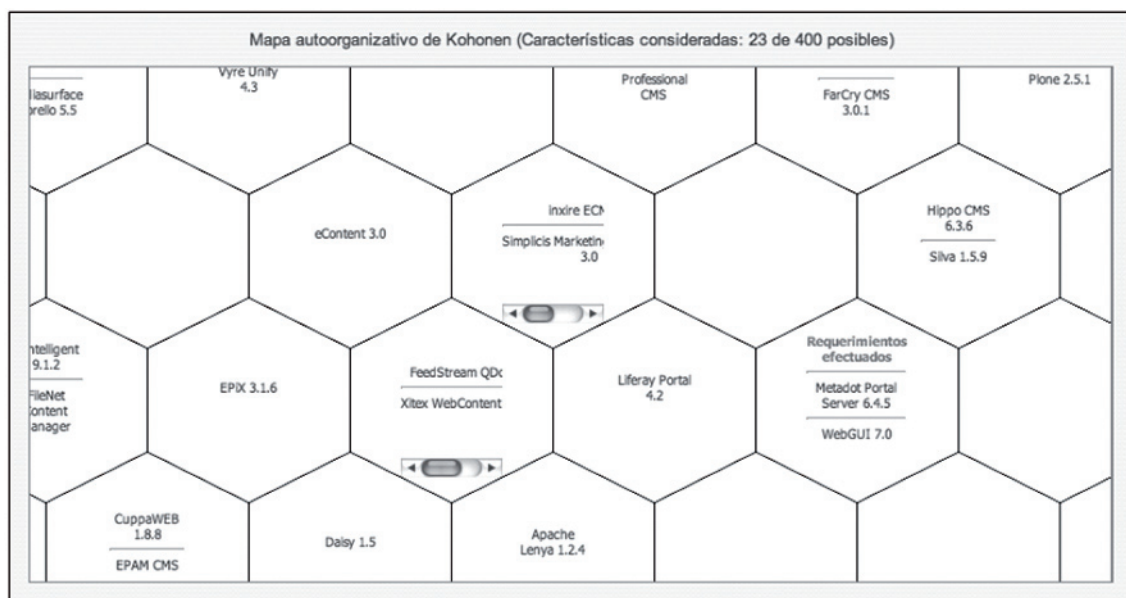


Fig. 6. Mapa autoorganizado obtenido en la cuarta prueba de CMS-SOM

También en esta prueba hubo problemas para ingresar los requerimientos en *CMS-Search*, *Produktfinder*, *CM-Overview* y *CMS-Matrix*. En este último, por ejemplo, fue necesario ingresar "mod_perl" en la casilla "Application Server", ya que ingresando "Perl" en la casilla "Programming Language" los resultados fueron insatisfactorios (en el campo correspondiente de los distintos CMS figuran valores con "Perl 5", "Perl 6", etc. y el sistema no es capaz de realizar una equiparación parcial). *Produktfinder*, *CM-Overview*, *CMS-Matrix* y *CMS-SOM* indicaron que el 0,6 ~ 1,2% de los CMS que constan en sus registros cumplieron con los criterios de búsqueda, por lo podría suponerse que esa es la participación de este tipo de CMS en el mercado. Por otro lado, el alto valor calculado para *CMS-Search* (12,3%) llevó a que se analizara la lista sugerida, y el resultado fue que la mayoría de esos CMS, a pesar de haber sido propuestos por el sistema, en realidad no cumple con los criterios de búsqueda.

En la quinta prueba, apareció solamente un CMS (Silva 1.5.9) en la celda de los requerimientos efectuados, y en una celda adyacente aparecieron cuatro más (CS EMMS Suite 4.3, Easy Publisher 1.8, icoya OpenContent 2.5.11 y Plone 2.5.1). Esto es correcto, pues sólo Silva 1.5.9 cumple completamente los requerimientos, ya que de los otros cuatro se desconoce si poseen la capacidad de generar en forma instantánea los documentos a publicar. También como parte de esta prueba se realizó la consulta de *CMS-Search*, *Produktfinder*, *CMOverview* y *CMS-Matrix* para comparar sus resultados con los de *CMS-SOM*. Es necesario aclarar que *Produktfinder* no permitió realizar la búsqueda necesaria para esta prueba, debido a la ausencia de Zope entre sus opciones, y que los valores de *CMS-Matrix* se obtuvieron ingresando "Zope" en la casilla "Application Server" y adicionalmente "Python" (el lenguaje usado por Zope) en la casilla "Programming Language", ya que, de lo contrario, la lista obtenida contenía demasiados valores incorrectos. La validez de las sugerencias de *CMSSOM* queda confirmada al observar que, de los 5 CMS propuestos por este sistema, dos (Plone 2.5.1 y Silva 1.5.9) aparecieron en al menos dos de los demás sistemas, otros dos (CS EMMS Suite 4.3 y Easy Publisher 1.8) aparecieron en al menos uno de los demás sistemas, y el restante (icoya OpenContent 2.5) está basado en Zope según consta en su propia página web.

En la sexta y última prueba de *CMS-SOM*, se ingresaron como preferencias los 400 valores que caracterizan al CMS Jahia. En el mapa autoorganizado obtenido, Jahia apareció correctamente en la celda de los requerimientos efectuados. Además, en la misma celda apareció Liferay Portal (12º en la tabla correspondiente, con 272 coincidencias), el cual es un CMS basado en la tecnología J2EE y más parecido a Jahia (también basado en J2EE) que, por ejemplo, TikiWiki (2º en la tabla, con 288 coincidencias), el cual está basado en PHP. Otros CMS basados en la tecnología J2EE (por ejemplo Magnolia y Oracle Universal Content Management) también aparecen en celdas próximas a la celda donde se encuentra Jahia, a pesar de tener un número menor de coincidencias en la tabla. Esta observación es muy importante: en el mapa es posible "ver" relaciones que en un *ranking* no son evidentes.

Conclusiones

Los responsables de la adopción de un CMS en una organización deben resolver el problema de identificar a los posibles proveedores de CMS, para poder realizar luego la selección del CMS que mejor cumple con los requerimientos de la organización. El problema es particularmente difícil porque, actualmente, el mercado de los CMS, "se encuentra en un estado de violenta agitación, con permanentes sacudidas, fusiones y adquisiciones. En los últimos años, la mayoría de las firmas que desarrollan CMS:

- ha sido adquirida por otra empresa o se ha fusionado con ella;
- salió del mercado, abandonando a sus clientes;
- ha estado ocupada adquiriendo otras compañías o productos para ampliar su oferta y parecer más integrada" (Doyle, 2004b).

Pero, además de las empresas, también los CMS han evolucionado en los últimos años, y con tanta rapidez que su selección se ha vuelto muy difícil, pues "incluso un analista de mercado de tiempo completo, pago para ser un experto en gestión de contenidos, será incapaz de conocer todos los productos y sus características técnicas" (Gilbane, 2003)

Ambas dificultades se presentaron también durante el desarrollo de *CMS-SOM*, ya que algunas empresas, cuyos productos habían sido elegidos para formar parte del conjunto predefinido de

CMS, fueron adquiridas por otras (por ejemplo, en octubre de 2006 *Hummingbird* fue adquirida por *Open Text* y *FileNet* pasó a manos de *IBM*, y en noviembre del mismo año *Oracle* compró *Stellent*), y la lista de características técnicas de los CMS también creció (por ejemplo, la compatibilidad con el sistema operativo Windows Vista debió ser agregada luego del lanzamiento de éste en enero de 2007).

A pesar de todo, el sistema *CMS-SOM* estuvo listo y funcionando a comienzos de 2008.

Las conclusiones más importantes de este proyecto son las siguientes:

- Se ha detectado que la mayoría de los expertos en CMS proponen gestionar la adquisición de un CMS a través de procesos como los que constituyen el área de conocimiento llamada *gestión de las adquisiciones del proyecto* en la "Guía de los Fundamentos de la Dirección de Proyectos" (PMBok Guide) del Project Management Institute.
- Se ha determinado que, de los procesos necesarios para gestionar la adquisición de un CMS, uno de los más problemáticos es la identificación de los proveedores de los CMS que mejor cumplen con los requerimientos de la organización.
- Se han estudiado los sistemas *CMS-Search*, *Produktfinder*, *CM-Overview* y *CMS-Matrix*, y se ha realizado una descripción crítica de los mismos.
- Se han identificado los CMS que son mencionados en, por lo menos, dos de los sistemas anteriores.
- Se ha establecido una lista de características descriptivas de los CMS, y se han recopilado definiciones y explicaciones para cada una de ellas.
- Se ha desarrollado el sistema *CMS-SOM*, como propuesta para identificar fácilmente los proveedores de los CMS que mejor cumplen con los requerimientos de una organización. Al ser instalado de forma adecuada, este sistema puede ser accesible a través de la Web.
- Los resultados experimentales han mostrado que el funcionamiento de *CMS-SOM* va mucho

más allá de realizar una simple tabla ordenada según las cantidades de coincidencias entre los requerimientos efectuados y los datos de los CMS, ya que en el orden espacial de los mapas son capturadas las "relaciones de familia" entre los CMS.

- *CMS-SOM* puede ayudar a reducir los costos que acarrea la selección de un CMS. Bob Doyle (2004a) menciona que (en los EE.UU.): *Los expertos de la industria como Bob Boiko, Tony Byrne, Jo Ann Hackos, Gerry McGovern y James Robertson y Ann Rockley pueden ser contratados como consultores neutrales por unos pocos miles de dólares para dictarle seminarios de un día de duración al personal clave que será el equipo de gestión de contenidos de la organización (...) Los informes de los analistas de la industria, donde se evalúa a las empresas que producen CMS y se reportan las tendencias tecnológicas en la gestión de contenidos y las herramientas que las implementan, oscilan entre unos pocos cientos de dólares y USD 1000 o más (...) Entre los documentos para el análisis de las necesidades y las solicitudes de propuestas se encuentran el CMS Planner (USD 300) de Boiko y el Requirements Toolkit (USD 550) de Robertson, que ofrecen plantillas para utilizar durante el proceso de selección.*

Agradecimientos

Al Dr. Ing. Raimundo O. D'Aquila, quien dirigió mi tesis de Maestría en Ingeniería en Sistemas de Información (de la cual este artículo es un resumen) y a la Lic. Adriana Almeida, por su inestimable ayuda en la preparación de los datos utilizados por *CMS-SOM*.

Referencias

- ANDERSON, E. (1990) *Choice Models for the Evaluation and Selection of Software Packages*. JMIS 6, 123
- BECHTOLSHEIM, M.; OBERBAUER, R. (2001) *Content Management ist Wissensmanagement - Strategie, Prozesse, Technologien*. IM 8, 7
- BOIKO, B. (2001) *Content Management Bible*. Wiley
- DAYHOFF, J. (1990) *Neural Network Architectures: an introduction*. Van Nostrand Reinhold
- DOYLE, B. (2004a) *Select a CMS in 15 Steps*. EContent , 7
- DOYLE, B., ED. (2004b) *Brief History of the CMS marketplace*. CMS Review
- DOYLE, B. (2005) *How many CMS are there?*. CMS-Forum
- DUJMOVIC, J. (1996) *A method for evaluation and selection of complex hardware and software systems*. San Francisco State University
- FONTELA, C. (2003) *Programación Orientada a Objetos - Técnicas Avanzadas de Programación*. Nueva Librería
- GILBANE, F. (2000), *What is Content Management?*. The Gilbane Report 10, 2
- GILBANE, F. (2003), *The Classification & Evaluation of Content Management Systems*. The Gilbane Report 3, 2
- HILERA, J.; MARTÍNEZ, V. (1995), *Redes Neuronales Artificiales: fundamentos, modelos y aplicaciones*. ra-ma
- HOWARD, J. (2003), *ECM - Don't buy it*. CMS Watch 3
- KOHONEN, T. (1996) *SOM_PAK: The Self-Organizing Map Program Package, Report A31*, Helsinki University of Technology
- KOHONEN, T. (2001) *Self-Organizing Maps*, 3ª Ed. Springer
- MICROSOFT CORPORATION (2004) *Ventajas y Beneficios de un Sistema de Gestión de Contenido Web*. Microsoft TechNet 6, 2
- NILSSON, N. (1998) *Artificial Intelligence: A New Syntesis*. Morgan Kaufmann
- PMI (PROJECT MANAGEMENT INSTITUTE) (2004) *Guía de los Fundamentos de la Dirección de Proyectos*, 3ª Ed.
- PRESSMAN, R. (2001) *Software engineering: a practitioner's approach*, 5ª Ed. McGraw-Hill
- ROBERTSON, J. (2002) *How to evaluate a content management system*. KM Column 1
- ROBERTSON, J. (2003a) *Why every small website needs a CMS*. CM Briefing 1
- ROBERTSON, J. (2003b) *So, what is a content management system?*. KM Column 6
- RÖWEKAMP, L. (2001) *Prinzipien und Aufbau eines Content Management Systems*. IM 8, 12
- SCRIVEN, M. (1991) *Evaluation Thesaurus*. Sage Publications
- STD (STEP TWO DESIGNS) (2004) *Content Management Requirements Toolkit*
- SUH, P. (2002) *Content Management Systems*. Glasshaus
- WELSTEAD, S. (1994) *Neural Network and Fuzzy Logic Applications in C/C++*. Wiley