

ANÁLISIS MULTIFUOCO DE FRAMEWORK PARA INTEGRACIÓN DE SERVICIOS EN SISTEMAS C4I

FRAMEWORK MULTI FOCUS ANALYSIS FOR INTEGRATION OF SERVICES IN C4I SYSTEMS

Edwin Juvenal Cedeño Herrera¹, Gloris Batista Mendoza¹, Raúl E. Dutari D.¹

¹Universidad de Panamá, Centro Regional Universitario de Veraguas. Facultad de Informática, Electrónica y Comunicación

edwin.cedenoh@up.ac.pa; gloris.batista@up.ac.pa; raul.dutari@up.ac.pa

RESUMEN

El Bus de Servicios Empresariales (ESB) es una infraestructura computacional distribuida utilizada en ambientes de gran escala y complejidad, como es el caso de departamentos de defensa, investigación y policía. Estos sistemas son fundamentales en la gestión de la información, la fusión de datos y la difusión, sobre todo en contextos de sistemas C4I (Comando, Control, Comunicaciones, Computadoras e Inteligencia). La elección adecuada de un producto que satisface todos sus requerimientos se convierte en un desafío debido a la magnitud y complejidad de estos sistemas. En esta investigación nosotros desarrollamos una evaluación mediante varios métodos, los cuales aplican distintos enfoques con el objetivo de determinar cuál es la mejor solución de bus de servicio ESB para entorno de sistemas C4I. Los productos considerados en esta investigación son el Oracle, Mule, Fuse y GlassFish. Los resultados muestran que Oracle ESB es la herramienta más robusta para entornos de sistemas C4I, con independencia del método y el enfoque del análisis.

PALABRAS CLAVE: Bus de Servicio Empresarial, C4I, Arquitectura Orientada al Servicio, Framework, Servicios Web.

ABSTRACT

Enterprise Service Bus (ESB) is a distributed computing infrastructure used in large-scale environments and complexity, such as departments of defense, research and police. These systems are fundamental in information management, data fusion, and dissemination, especially in C4I system contexts (Command, Control, Communications, Computers, and Intelligence). Choosing the right product that meets all your requirements becomes a challenge due to the magnitude and complexity of these systems. This paper discusses different solutions SOA service bus applied to such environments. This research develops an evaluation through several methods, which apply different approaches in order to determine which is the best ESB service bus solution for C4I systems environment. The products considered in this research are Oracle, Mule, Fuse and GlassFish. The results show that Oracle ESB is the stronger tool for C4I systems environments, regardless of the method and approach of the analysis.

KEYWORDS: Enterprise Bus Service, C4I, Service Oriented Architecture, Framework, Web Services.

Artículo recibido: 12 de enero, 2020

Artículo aceptado: 28 de marzo, 2020

INTRODUCCIÓN

En las grandes compañías, departamentos militares, departamentos de administración pública, etc. los líderes son cada vez más presionados por la toma de decisiones y sus procesos estratégicos de negocios. ¿Qué proceso es estratégico para una determinada entidad pública o privada?, puede variar significativamente dependiendo del área a la que está dedicada esa compañía o departamento, pero un factor común es que los gerentes quieren que sus departamentos de Tecnología de Información mejoren de forma considerable el flujo de información y datos que conducen a tomar decisiones claves. No importa si son las Fuerzas Armadas tratando de integrar todos sus recursos, o un proveedor de materiales de construcción luchando para ordenar el flujo en una compleja cadena de distribución; siempre hay significativos desafíos técnicos a ser superados (Alghamdi, Nasir, Iftikhar, & Nafjan, 2010). La información es retenida, oculta en aplicaciones dentro de diferentes departamentos u organizaciones, y cuesta tiempo y dinero tratar de buscar información entre esos datos sueltos.

En los últimos años se han venido presentado varias tendencias tecnológicas como Service Oriented Architecture (SOA), Enterprise Application Integration (EAI), Business-to-Business (B2B) y Servicios Web (Wu & Tao, 2010). Estas tecnologías intentan direccionar los desafíos de mejorar los resultados en la toma de decisiones e incrementar el valor de los Procesos Integrados de Negocio, y han logrado atraer la atención de líderes de IT, vendedores, y analistas de las industrias. Enterprise Service Bus (ESB) toma las mejores características de estas y otras tendencias tecnológicas, y propone una solución global.

El concepto de ESB es un nuevo acercamiento a la integración que puede proporcionar las bases para una débilmente acoplada, y altamente distribuida red de integración que puede escalar más allá de los límites de un Framework EAI estructurado en modo hub-and-spoke (Wu & Tao, 2010).

En este trabajo analizaremos algunos productos de integración con ESB, que pueden ser utilizados en entornos sumamente críticos como son los sistemas de Command, Control, Communications and Intelligence (C4I) (Alghamdi & Ahmad, 2010), y conectar las diferentes áreas, como por ejemplo: los departamentos de las fuerzas armadas para compartir la información necesaria en la toma de decisiones en momentos cruciales de combate. Pero antes de realizar este análisis, se expone como un Bus de Servicios es una parte fundamental en una SOA, proporcionando servicios débilmente acoplados, y facilitando además un servicio seguro de transferencia de mensajes entre aplicaciones e interoperabilidad usando servicios web y tecnologías relacionadas (Alghamdi, Nasir, Iftikhar, & Nafjan, 2010). El objetivo de este análisis de herramientas de ESB es determinar cuál es la mejor alternativa de integración de servicios en entornos de sistemas C4I, para ello se utilizarán varios métodos de comparación, considerando soluciones representativas del sector privativo y de código abierto.

Un ESB es una plataforma de integración basada en estándares, que combina varias funcionalidades para conectar, de forma confiable, y coordinar la integración de un número considerable de diferentes aplicaciones a través entornos empresariales con integridad transaccional (Goudarzi, 2007).

Cada desarrollador construye su ESB basado en sus propios recursos y puntos de vista, por esto no existe un consenso generalizado de lo que debe y no debe tener una solución de ESB. Sin

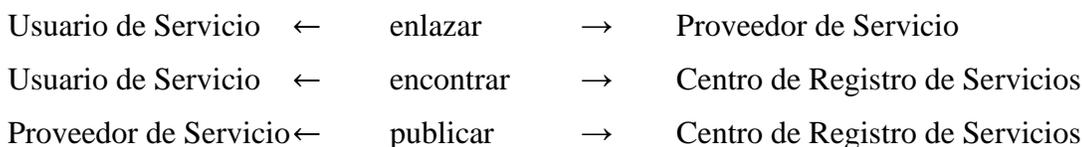
embargo, sí existe un consenso casi generalizado de las funciones principales que debe tener un ESB, las cuales son: transmisión de mensajes, servicio web, transformación de datos y ruteo inteligente.

Actualmente existe una gran variedad de soluciones ESB en el mercado, en la Tabla N° 1 podemos ver algunas soluciones de código abierto con sus respectivos desarrolladores (MuleSoft, INC., 2017), (Red Hat Inc., 2016), (GlassFish, 2017), (Bayer, 2009). Todas estas soluciones se basan en SOA, la cual es un modelado de componentes que puede relacionar las diferentes unidades funcionales (llamadas servicio) de aplicaciones, a través de las interfaces y conexiones entre servicios. La interface que es definida de una forma neutral, permite crear la interacción entre diferentes servicios de una manera uniforme y transparente.

Tabla N° 1: Soluciones ESB Open Source

Solución ESB	Desarrollador
Mule	Proyecto Mule (Open Source)
Fuse	Apache ServiceMix Committers (Open Source)
GlassFish	Sun Microsystems (Open Source)
OpenESB	Sun Microsystems (Open Source)

El modelo básico de la arquitectura SOA tiene tres roles o actores detallados así: “Proveedor de Servicio”, “Usuario de Servicio”, y “Centro de Registro de Servicios”; y también incluye tres acciones correspondientes a las interacciones entre los componentes, las cuales se mencionan a continuación, “publicar”, “encontrar”, “enlazar (e Invocar)” (Wu & Tao, 2010). Las interacciones entre los roles se pueden llevar a cabo de la siguiente manera:



Estructura Básica de ESB

El Bus de Servicios Empresariales es un modelo de arquitectura que puede soportar interacciones de participantes virtuales de comunicación y administrar sus servicios. Este proporciona la conexión entre el servicio proveedor y el servicio solicitante, incluso si ellos no están perfectamente conectados, este también puede hacer que interactúen. Este modelo puede ser implementado con una variedad de tecnologías de middleware y modelos de programación.

En esta arquitectura el solicitante de servicio no necesita conocer la implementación física del proveedor de servicio en el modelo de arquitectura ESB, ya que este es responsable del reparto del mensaje solicitante al correspondiente proveedor de servicio. De la misma forma, los proveedores de servicio tampoco necesitan la fuente del mensaje solicitante cuando ellos responden los mensajes que han estado recibiendo. ESB, en sí mismo, no es visible ni al solicitante ni al proveedor del servicio. Esta arquitectura está principalmente compuesta de una Pasarela de Servicio, un Adaptador de Servicio, y un Registro de Servicio. La Pasarela de servicio es la parte central del ESB y es responsable de los servicios de ruteo y conmutación; el Adaptador de Servicio es usado para implementar la conversión de protocolo de los servicios e información; en tanto que el Registro de Servicio ofrece ayuda para el Ruteo de Servicio (Wu & Tao, 2010).

Los sistemas C4I, han sido llamados de esta manera por sus características de “Command”, que es la autoridad que un líder ejecuta sobre sus subordinados en virtud de su rango asignado; “Control”, la autoridad que puede ser menos que la orden directa del líder y es ejecutada sobre las actividades o subordinados de otra organización; mientras que “Computers and Communications” por la actividades de proceso y transporte de información; y finalmente “Intelligence” para hacer referencia a la información y conocimiento obtenidos a través de observación, investigación, análisis, o entendimiento (Alghamdi & Ahmad, 2010).

Actualmente los Sistemas C4I son usados en varios departamentos militares y civiles tales como departamento de defensa, policía, investigación, caminos, trenes, aeropuertos, aceite y gas donde existen los escenarios de control y comando. Sin embargo, el principal enfoque de estos sistemas es en aplicaciones de defensa.

Los sistemas C4I están conformados por personas, procedimiento, tecnología, doctrina y autoridad, y juegan un creciente rol en la administración de información, fusión de datos, y disseminación (Alghamdi, Ahmad, & Nasir, 2010).

Los procesos de la tecnología C4I deben acelerar los establecimientos de enlaces entre los diferentes departamentos que contienen la información necesaria y a través de estos enlaces deben proporcionar el acceso inmediato a las armas militares.

Para interconectar diferentes sistemas de defensa para un adecuado intercambio de información especialmente durante una guerra, ESB juega un rol importante. En este escenario los sistemas de defensa requieren sistemas débilmente acoplados que puedan trabajar juntos y también en forma independiente. Es decir, los sistemas de defensa requieren un middleware que pueda interconectar sus sistemas entre sí, el cual debería ser confiable y fuerte en interoperabilidad y transferencia de datos (Alghamdi, Nasir, Iftikhar, & Nafjan, 2010).

Una de las más importantes consideraciones en un sistema C4I, es la selección del Framework apropiado debido a que la arquitectura de este es de vital importancia en el diseño y desarrollo de los sistemas de información. Los Frameworks de arquitectura proporcionan aproximaciones sistemáticas al desarrollo de la arquitectura (Alghamdi & Ahmad, 2010).

En este trabajo se examinan cuatro herramientas, por considerarse que son las que mejor se adaptan a los sistemas C4I, según trabajos anteriores (Alghamdi, Ahmad, & Nasir, 2010). Estas herramientas las podemos clasificar en privativas y de código abierto. Las de código abierto son: Mule ESB, Fuse ESB y GlassFish, por parte de las privativas se incluye a Oracle ESB.

Oracle ESB, este producto de Oracle ha sido desarrollado bajo el estándar de Business Integration (BI) y enfocado hacia Business Process Management (BPEL), ESB, Enterprise Messaging Service (EMS), Oracle Data Hub (ODH), RFID y Sensor (SES), Partner Integration (B2B), Enterprise Connectivity (Adapters) Business Activity Monitoring (BAM) y Service Oriented Architecture (SOA).

Las características de interoperabilidad e integración están enfocadas al ambiente de desarrollo integrado basado en Eclipse (IDE), un browser basado en herramientas para que los usuarios puedan crear composición de aplicaciones a través de procesos de orquestación, diseño con tiempo de integración rápido, soporte y gestión del ciclo de vida SOA, soporte para OSB en tiempo de ejecución y administración de procesos en tiempo de ejecución.

Por otra parte, en cuanto las características de alta disponibilidad, Oracle ESB ofrece los siguientes productos, SOA Suite High Availability, BPEL Process Manager High Availability, Oracle WSM Configuration on Clustered Environment, ONS Topology, Synchronous and Asynchronous Node Connectivity, Load Balancer, Multiple Domain Clustering, Dehydration Store Database, Automatic Nodes Propagation Change Configuration.

Mule ESB, es también denominado MuleSource, y hoy día se conoce como MuleSoft, ofrece un modelo de desarrollo simple y una arquitectura liviana para integrar, inter operar y crear servicios de forma fácil y rápida. Mule ESB requiere pocos recursos de CPU y memoria, simplificando de esta forma el despliegue y mantenimiento de las soluciones.

Dentro de las características de seguridad cuenta con acceso a los mensajes a través de CXF, acceso al contexto usando WS-Security, Web Services seguros con conectores WSS4J, validación de firmas, auditoría del log, acceso al código fuente, una comunidad de extensiones y foros de seguridad. Otras características importantes son integración y alta disponibilidad, y para hacer frente a estos requerimientos cuenta con una consola de gestión, alta disponibilidad y capacidades de recuperación ante fallos, SEDA servicios de encolado de eventos, encolado de eventos en memoria, alta disponibilidad para HTTP, JMS, WebSphere MQ, JDBC, FILE, FTP, Clustered, adicionalmente, cuenta con servicio de registro integrado, transacciones multi recursos, conectores WebSphere MQ, conectores JDBC, soporte técnico, SLA's Empresariales, guía de usuarios y manuales de calidad comercial, además de artículos de conocimientos básicos.

Mule provee conectividad de inmediato, es decir, conectar y listo, también ofrece transportes comunes como JMS, HTTP, SMTP, FTP, POP3, y los XMPP son soportados de forma nativa, así como los webs services. El sistema de mensajes que utiliza típicamente Mule ESB es JMS, sin embargo, otros servidores de mensajes pueden ser implementados, como por ejemplo MS-Messaging Queuing (MSMQ), IBM WebSphere MQ o Tibco Rendezvous. Mule ESB no requiere un código de programación específico de una Interface de Programación de Aplicaciones para ejecutar los componentes. Provee soporte de integración con Spring Framework y con la gestión de procesos del negocio (BPM).

En Mule ESB, cuando las aplicaciones se conectan y quieren compartir datos con otras aplicaciones, estas leen los datos de una forma, luego los cambian completamente como sea necesario para que estos puedan ser leídos por otras aplicaciones, con esta funcionalidad Mule ESB permite la integración de todo tipo de aplicaciones que no fueron construidas pensando en la integración.

Fuse ESB, puede ser embebido en los hosts finales fácilmente, esto permite interactuar con sistemas distribuidos inteligentes, sin un servidor central de gestión. Fuse ESB posee una arquitectura de conexión inmediata y trabaja con otros componentes de integración que ya se están utilizando (OSGi, JMS, JCA y JMX). Fuse ESB está basado en Apache ServiceMix y está totalmente basado en estándares de código abierto.

En Fuse no hay un servidor central de gestión, esto permite que los sistemas distribuidos interactúen entre sí de forma inteligente. Este también provee soporte para Spring Framework, ofreciendo esto un contenedor liviano para componentes de aplicación. Fuse ESB también soporta interoperabilidad a través Web Services complejos y servicios distribuidos o servicios independientes. El contenedor JBI soporta componentes que incluyen a Normalized Message

Router (NMR), para localizar el proveedor de servicios apropiado. NMR provee una interface para la conectividad entre el proveedor de servicios y conexiones de muchos a muchos.

Fuse ESB implementa la interoperabilidad y alta disponibilidad con sus características de ACTIVE & STANDBY en ServiceMix, Clustering en ServiceMix, Separate Host, Distributed Message Routing, despliegue dinámico, desarrollo rápido, interface de mensajería estándar e integración de envío directo.

GlassFish, provee una plataforma de integración liviana, para un desarrollo rápido de herramientas y despliegue de componentes SOA libre de dependencias y con mucha flexibilidad. GlassFish ESB es fácil de integrar y proporciona interoperabilidad. Este ESB incluye un servidor de aplicaciones, herramientas como NetBeans, JBI para despliegue de aplicaciones en tiempo real, ingeniería de integración, adaptadores para sistemas externos y un instalador simple. También el contenedor JBI incluye soporte para componentes e incluye a NMR, para localizar el proveedor de servicios adecuado (Alghamdi, Nasir, Iftikhar, & Nafjan, 2010).

GlassFish ESB está basado en Open ESB que proporciona en una plataforma de integración, un Enterprise Architecture Integration (EAI) y Service Oriented Architecture (SOA). GlassFish está apoyado en un gran número de estándares, como JBI, Java EE, SOAP y otros más, esto permite un desarrollo flexible, soluciones adecuadas para la integración de los sistemas con un gran número de componentes, incluyendo componentes de enlace (adaptadores) y motores de servicio (procesadores). GlassFish ESB utiliza componentes de JBI que proporcionan arquitecturas asíncronas y diseño de modelos desacoplados, esto permiten escalabilidad vertical y horizontal. GlassFish permite tomar ventaja de Staged Event-Driven Architecture (SEDA), debido a la asíncrona naturaleza de los mensajes, esto permite minimizar los bloqueos de hilos, asociados a los requerimientos de memoria y escalabilidad de aplicaciones sin código específico.

MATERIALES Y MÉTODOS

Las herramientas o productos descritos en la sección anterior son consideradas representativas en cada uno de los segmentos (privativas y código abierto), y como se puede observar en la Figura N° 1(a y b), las cuales corresponde a estudios realizados por las consultoras Gartner y Forrester respectivamente, queda de manifiesto que las mismas son herramientas referentes en integración y ESB.

En el cuadrante mágico para infraestructura de aplicaciones para proyectos de integración sistemática de aplicaciones, que se muestra en la Figura N° 1(a), podemos observar el comportamiento de los productos ESB en el mercado. En este caso podemos observar que MuleSoft, se encontraba en el cuadrante del nicho de jugadores, una posición modesta comparada con Oracle que ya para ese tiempo se encontraba de puntero en el cuadrante de los líderes. Esta situación no parece muy alarmante considerando que Oracle es privativa y MuleSoft es de código abierto, con lo cual la conducta del mercado está basada en la confianza sobre herramientas que tengan respaldo de compañías grandes, como es el caso de Oracle.

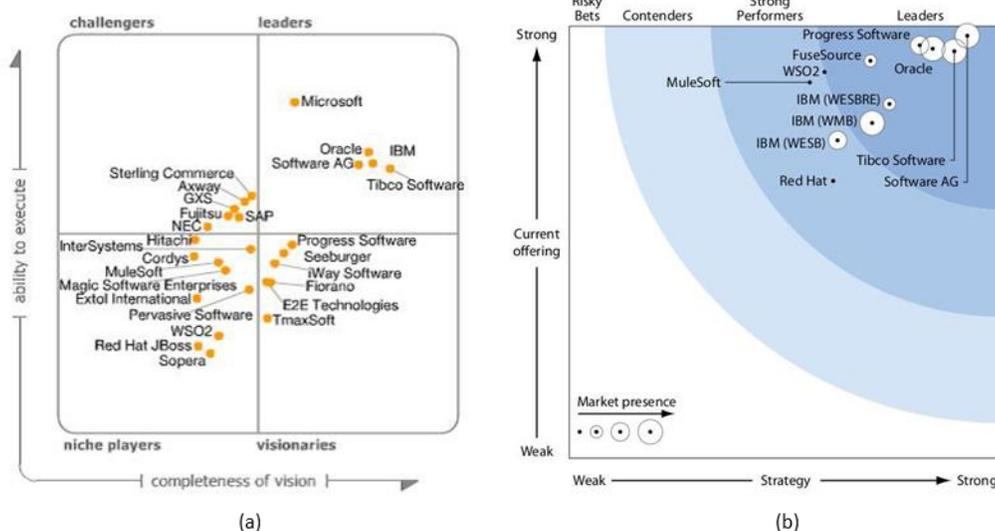


Figura N° 1: (a) Cuadrante Mágico para Infraestructura de Aplicaciones de Integración, Gartner – 2010. Fuente: (Thompson, et al., 2010). (b) Forrester Wave Enterprise Service Bus, Q2-2011. Fuente: (Vollmer, Gilpin, & Rose, 2011)

Por otra parte, se muestra la ola de Forrester, para los Enterprise Service Bus (ESB) la cual se puede observar en la Figura N° 1(b), donde se nota el posicionamiento de los productos sujeto de análisis de una forma más concreta, debido a que este estudio es más específico que el anterior en cuanto al tema de los ESB. Forrester muestra que MuleSoft se encuentra en el espectro de los productos de fuerte rendimiento, por su parte FuseSource se encuentra mejor posicionado que el anterior al estar en el espectro de los líderes, y finalmente encontramos a Oracle en mejor posición que los anteriores, en el espectro de los líderes, pero con mayor fortaleza que FuseSource.

De ambos estudios se puede extraer que el mercado está utilizando en su mayoría Oracle que es un producto propietario, sin embargo, los productos de código abierto no están tan lejos de este último, es decir, que también están ofreciendo buenos productos como soluciones a los ESB en general.

Realizar un análisis concreto sobre un conjunto de herramientas, con el objetivo de determinar cuál de ellas es la mejor alternativa y, además, en un contexto particular y con requerimientos específicos, no es una tarea trivial. En primera instancia los estudios anteriores realizados al respecto de los ESB para sistemas C4I, como era de esperarse, tienen diferentes enfoques, métodos de comparación y criterios, por lo tanto, en este trabajo presentamos un análisis de algunos métodos de comparación, con el objetivo de identificar cual sería la herramienta ESB más adecuada para entornos C4I.

El primer método de análisis está basado en los sistemas C4I que han pasado de entornos militares, a ser utilizados en el entorno civil, policial, investigación, carreteras, ferrocarriles, aeropuerto, oleoductos, gaseoductos y otros. Además, hay que considerar el hecho de que dichos sistemas están compuestos por personas, procedimientos, tecnologías, jerarquías de autoridad, que juegan un papel importante en la gestión, integración y comunicación de la información. Por todo lo planteado anteriormente, hay dos características que destacan ante las demás, y que se requieren

en los sistemas C4I, una de ellas es la seguridad de mensajería y la otra es, por supuesto, la interoperabilidad.

Multi-Criterial Decision Making Techn (MCDM)

En cuanto al primer método utilizado para la evaluación proponemos el MCDM, el cual consiste en un conjunto de pasos que se muestran a continuación.

1. Selección del objetivo
2. Criterios de decisión
 - Interoperabilidad
 - Extensibilidad
 - Mensajería
 - Facilidad
 - Disponibilidad
3. Determinar alternativas
 - GlassFish
 - Fuse
 - Mule
4. Definir jerarquía
5. Asignación de Prioridades
6. Cálculos de los pesos
7. Test de consistencia

En primer paso consiste en el establecimiento del objetivo, luego se definen los criterios a considerar en la evaluación, en el paso tres se eligen las alternativas (productos) que serán evaluadas, luego se define una jerarquía, posteriormente en el paso cinco se asignan las prioridades a los criterios, en el paso seis se calculan los pesos y finalmente en el paso siete se verifica la consistencia del test, el resultado de este último debe mantenerse menor a 10% para que la evaluación sea aceptable.

La construcción de la jerarquía consiste en ordenar el objetivo general consecuentemente con los criterios que permitirán lograrlo, de igual forma cada criterio puede subdividirse en sub-criterios, en cuyo caso deben ser ordenados en el nivel inmediatamente inferior a los criterios principales como puede observar en la Figura N° 2. La interoperabilidad se subdivide en otros sub-criterios: red, semántica y sintaxis. De igual forma la mensajería se subdivide en velocidad, seguridad y confiabilidad. El último criterio que se subdivide es disponibilidad, que estará formado por los sub-criterios, soporte a fallos, sin estado y estado completo.

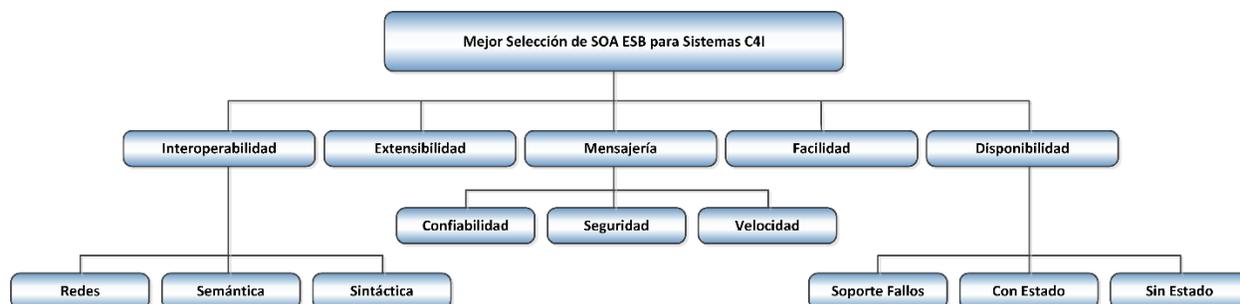


Figura N° 2: Jerarquía de MCDM

El siguiente paso es la asignación de prioridades que está basada en información de los trabajos previos (Alghamdi, Ahmad, & Nasir, 2010), se utiliza entonces la información de la Tabla N° 2.

Tabla N° 2: Definición de Prioridades en MCDM

Intensidad	Definición
1	Igual importancia
2	Poca importancia
3	Moderada importancia
4	Moderada importancia +
5	Fuerte importancia
6	Fuerte importancia +
7	Muy fuerte importancia
8	Muy fuerte importancia +
9	Extremadamente importante

El cálculo de los pesos de cada nodo está en función de las prioridades asignadas según la Figura N° 3. Estos pesos son también importados de trabajos anteriores que han sido consensuados (Alghamdi, Ahmad, & Nasir, 2010). El esquema representa la jerarquía de pesos para los criterios y sub-criterios, estos pesos se especifican en un ámbito local, es decir, en base al sub-criterio particular que se está analizando, o en un ámbito global, que se refiere al peso correspondiente en términos de los criterios principales.

Los criterios utilizados para la evaluación juegan un papel fundamental, en términos de los requerimientos, ya que los criterios serán más importantes en la medida que ellos satisfagan los requerimientos relevantes, por lo tanto, es importante saber los pesos que se han asignado a cada criterio, y si estos son cónsonos con los requerimientos que se buscan, de esta forma podemos tener un mayor grado de certeza en cuanto a los resultados de la evaluación de las herramientas, para tal efecto si se realiza una ranking de criterios, podemos verificar que el criterio más importante en la evaluación es la interoperabilidad, que en el fondo, como se explicó anteriormente en el enfoque para sistemas C4I, es uno de los más importantes.

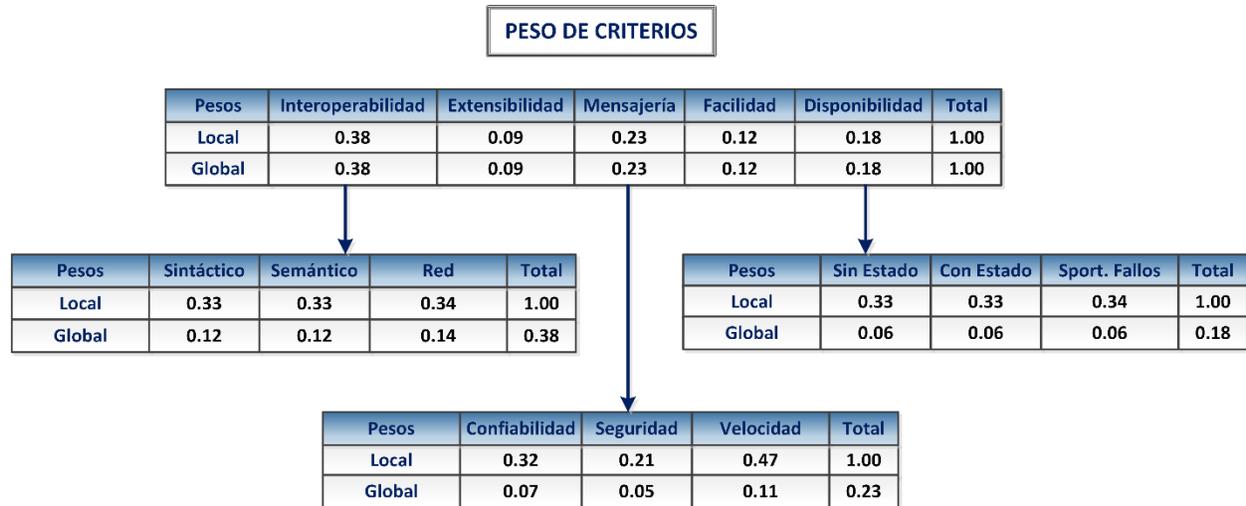


Figura N° 3: Asignación de Pesos a los Criterios y Sub-Criterios en MCDM

Finalmente, se presentan los resultados de la evaluación que se muestran en la Figura N° 4, en esta se puede observar que de las herramientas de código abierto, Fuse, Mule y GlassFish, la que ha obtenido mejor puntuación ha sido Mule, seguida de GlassFish y finalmente aparece Fuse. Hay que tener cuidado al interpretar los resultados ya que en el caso de que el criterio mayor valorado para un entorno de C4I hubiese sido la extensibilidad, la mejor herramienta hubiese sido Fuse, ya que obtuvo mejores puntuaciones que Mule en ese criterio.

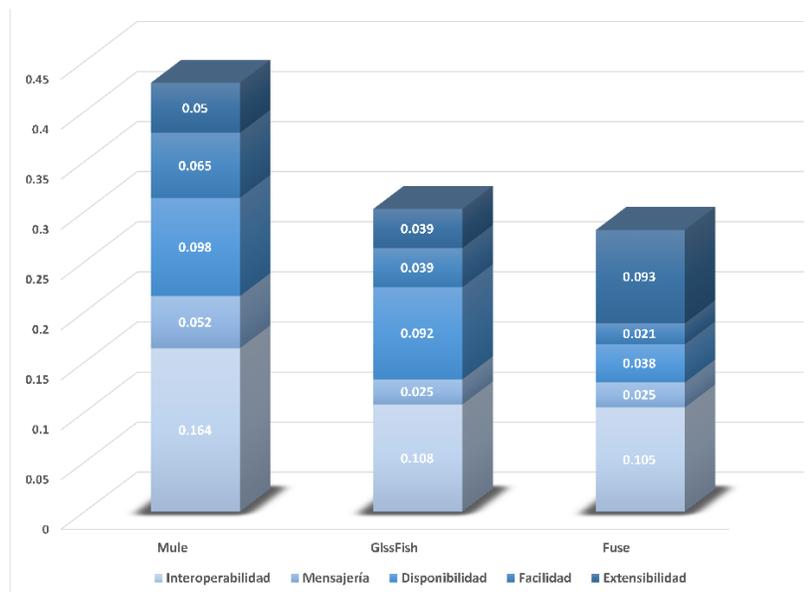


Figura N° 4: Resultados de la Evaluación de los ESB

La segunda parte de este análisis se concentra en los trabajos enfocados hacia las arquitecturas de defensa (C4I), en donde éstas son consideradas cruciales en comparación con otras arquitecturas empresariales. Estos sistemas están basados en la interoperabilidad de los diferentes dominios militares, aire, mar y tierra. Esta necesidad manifiesta de una alta interoperabilidad,

seguridad y eficiencia para los sistemas C4I, determina los factores de seguridad, interoperabilidad y disponibilidad, como los más relevantes para la evaluación.

Analytical Hierarchy Process Method (AHP)

El AHP se sugiere como un método analítico de procesos jerárquicos, este método se adapta muy bien cuando las decisiones tienen que ser tomadas basadas en numerosas y distribuidas evaluaciones (Siddiqui, Abdullah, & Khan, 2011). Este método inicia con el establecimiento del objetivo en la parte superior de la jerarquía, posteriormente la jerarquía se descompone por niveles, en el segundo nivel se establecen las herramientas objeto de la evaluación, luego en el tercer nivel, para cada nodo de nivel dos, se establecen los criterios con estos serán analizados.

Los ESB pueden ser comparados utilizando un método denominado PairWise, que consiste en realizar comparaciones por parejas, y asignando pesos según unos niveles de intensidad que se muestran en la Tabla N° 3.

Tabla N° 3: Niveles de Intensidad en AHP

Intensidad	Definición
1	Idéntico
3	Considerablemente a favor
5	Fuertemente soportado
7	Muy fuertemente soportado
9	Drásticamente a favor

Luego de realizar las comparaciones respectivas entre todas las alternativas sujeto de evaluación, en este caso se obtienen los resultados que se pueden observar en la Figura N° 5.

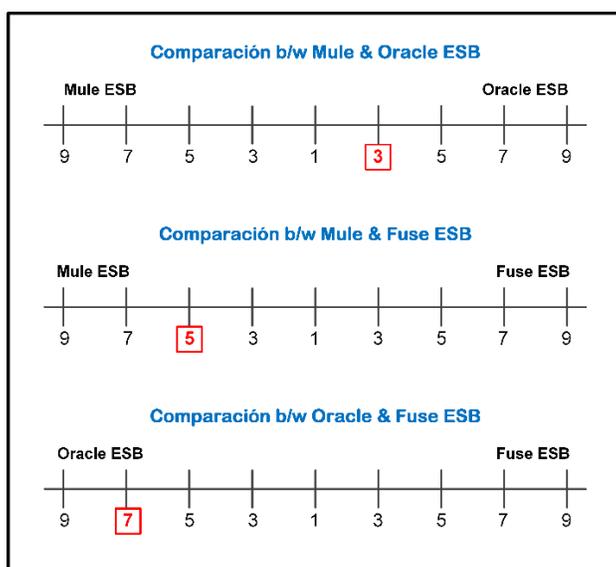


Figura N° 5: Comparación PairWise Oracle & Fuse & Mule. Fuente: (Siddiqui, Abdullah, & Khan, 2011)

Posteriormente se procede a construir la matriz de comparación, en nuestro caso será de 3x3, en esta la diagonal principal de la matriz será igual a uno, luego los lugares superiores a la diagonal principal se rellenan con los valores actuales si el resultado de la premisa de comparación está más

a la izquierda, en lado contrario se colocará el valor recíproco, de esta forma se obtiene la matriz, que se puede observar a continuación:

$$\begin{array}{l} \text{Mule} \\ \text{Oracle} \\ \text{Fuse} \end{array} \begin{bmatrix} 1 & 1/3 & 5 \\ 3 & 1 & 7 \\ 1/5 & 1/7 & 1 \end{bmatrix}$$

Para calcular el vector de prioridad, utilizamos el “Eigenvector”, el cual se calcula sumando los términos de cada columna, con los cual se obtiene lo siguiente:

$$X = \begin{bmatrix} 1 & 1/3 & 5 \\ 3 & 1 & 7 \\ 1/5 & 1/7 & 1 \end{bmatrix}$$

$$\sum = [21/5 \quad 31/21 \quad 13]$$

Seguidamente se dividen todos los elementos de la matriz entre la sumatoria de su columna correspondiente, después de la división tendremos los pesos relativos normalizados, de modo que si se suma cada columna deberá ser igual a uno, así:

$$X \equiv \begin{bmatrix} 5/21 & 7/31 & 1/13 \\ 15/21 & 21/31 & 7/13 \\ 1/21 & 3/31 & 1/13 \end{bmatrix}$$

$$\sum = [1 \quad 1 \quad 1]$$

Para asegurar que el vector resultante esté normalizado, hay que dividir todos los valores por filas, entre el promedio de la suma de los valores por columna entre el número de valores por filas, de esta forma se tiene:

$$\equiv 1/3 \begin{bmatrix} 0.8484 \\ 1.93 \\ 0.2212 \end{bmatrix} \text{ OR } \equiv \begin{bmatrix} 0.8484/3 \\ 1.93/3 \\ 0.2212/3 \end{bmatrix}$$

Finalmente se obtiene el “Eigenvector”, en el cual la suma de sus elementos debe ser igual a uno, como se puede observar a continuación:

$$\begin{array}{l} \text{Mule} \\ \text{Oracle} \\ \text{Fuse} \end{array} \begin{bmatrix} 0.2828 \\ \mathbf{0.6434} \\ 0.0738 \end{bmatrix}$$

De esta forma se puede constatar que Oracle es la mejor opción para satisfacer los requerimientos de sistemas C4I, en comparación con las demás herramientas analizadas. Es importante señalar que una vez se tenían las evaluaciones PairWise, se pudo inferir por lógica que, si Oracle es mejor que Mule, y Mule es mejor que Fuse, entonces Oracle es mejor que Fuse, sin embargo, estas mediciones no permiten establecer razones como lo permite el “Eigenvector”, es decir, establecer razones de cuantas veces es mejor uno de otro.

RESULTADOS

Una vez que se han llevado a cabo ambos estudios se puede decir que, entre las herramientas de código abierto analizadas, la mejor opción es Mule, por otro lado, las de código abierto comparadas con las herramientas privativas, la mejor valorada es Oracle ESB.

Hay que señalar que los criterios utilizados son bastante similares, con lo cual pueden ser considerados para próximos estudios. Por otra parte, los métodos utilizados son diferentes, sin embargo, los enfoque a pesar de considerar los sistemas C4I, el primero contempla el hecho de utilizarlo en otros contextos, en cuyo caso pueden variar los requerimientos, en tal caso la evaluación podría tener sesgos.

En la Tabla N° 4, se puede observar un resumen del análisis realizado, en donde resalta el hecho de que el contexto, el enfoque y el origen de las herramientas son factores fundamentales para llegar a conclusiones válidas en este tipo de estudio.

Tabla N° 4: Resumen de la Evaluación de los ESB

Método de evaluación	Contexto	Enfoque	Producto mejor valorado
Forrester	Global	Generalista	Oracle
Gartner	Global	Generalista	Oracle
MCDM	C4I y otros	Seguridad de Mensajería Interoperabilidad	Mule
AHP	C4I	Seguridad Interoperabilidad Disponibilidad	Oracle

DISCUSIÓN DE LOS RESULTADOS

Estos resultados están condicionados a las herramientas que se han considerado, sin embargo, como hemos mencionado antes, las que se han incluido en el estudio es porque hemos considerado que son las más representativas de los segmentos de herramientas privativas y las de código abierto, información que se reafirma con los análisis de Forrester Wave y Gartner en su Magic Quadrant Research Methodology.

Los resultados de las evaluaciones de los métodos de análisis de las herramientas para ESB en entornos C4I, que se muestran en la Tabla N° 4, indican que independientemente del enfoque y del contexto las herramienta que mejores características y rendimiento ofrece para estos entorno es Oracle.

Hay que dejar claro que en la aplicación del método MCDM, no se incluyó a Oracle entre los ESB que se analizaron, en este caso la herramienta recomendada es Mule. Este ESB es el mejor valorado para las herramientas de código abierto, por lo que consideramos que es su mayor exponente. Sin embargo, en el análisis con el método de AHP si fueron incluidas Oracle y Mule, además de Fuse. Para los análisis con los métodos AHP y MCDM, se utilizaron en común los criterios de seguridad e interoperabilidad como criterios con mayor ponderación, esto nos permite inferir que si repetimos el experimento incluyendo a Oracle, el producto mejor valorado sería Oracle.

CONCLUSIONES

Los ESB ofrecen una alternativa de solución al problema de la integración entre los diferentes sistemas que conforman los sistemas C4I, además de que satisfacen sus requerimientos.

Los métodos de evaluación ofrecen formas de cuantificar en función del enfoque de los mismos, para decidir basados en criterios objetivos y de una manera más científica, sobre la elección de un producto en particular destinado al entorno de los sistemas C4I.

La selección de los criterios para la evaluación de un ESB está condicionada por el contexto en donde se despliega la tecnología, ya que es mandatorio considerarlo durante el proceso de evaluación.

Según las cuatro referencias de evaluación utilizadas en esta investigación hemos concluido que la herramienta más robusta para los sistemas C4I es Oracle, independientemente del contexto y del enfoque del análisis.

REFERENCIAS

- Alghamdi, A. S., Ahmad, I., & Nasir, M. (2010). Selecting the best alternative SOA service bus for C4I systems using multi-criteria decision making technique. *2010 IEEE Region 8 International Conference on Computational Technologies in Electrical and Electronics Engineering (SIBIRCON)*, 790-795. Recuperado de: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5555084&isnumber=5555004>
- Alghamdi, A., & Ahmad, I. (2010). Comparative Analysis of Defense Industry Frameworks for C4I System. *Second International Conference on Computer Engineering and Applications*, 443-447.
- Alghamdi, A., Nasir, M., Iftikhar, A., & Nafjan, K. (2010). An interoperability study of ESB for C4I systems. *International Symposium on Information Technology*, 733-738.
- Bayer, T. (2009). *OpenESB and ServiceMix in Comparison*. Retrieved 07 19, 2017. Recuperado de: <https://www.predic8.com/openesb-servicemix-comparison.htm>
- GlassFish. (2017). *GlassFish: The Open Source Java EE Reference Implementation*. Recuperado de: <https://javaee.github.io/glassfish/>
- Goudarzi, N. (2007, 08). *Enterprise Service Bus*. Recuperado de: <http://www.javadev.org/files/Enterprise%20Service%20Bus.pdf>
- MuleSoft, INC. (2017). *mule esb community*. Recuperado de: <https://developer.mulesoft.com/>
- Red Hat Inc. (2016). *Red Hat JBoss Fuse*. Recuperado de: <https://developers.redhat.com/products/fuse/overview/>
- Siddiqui, Z., Abdullah, A., & Khan, M. (2011). Qualified Analysis b/w ESB(s) Using Analytical Hierarchy Process (AHP) Method. *2011 Second International Conference on Intelligent Systems, Modelling and Simulation*, 100-104. Recuperado de: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5730328&isnumber=5730309>

Thompson, J., Lheureux, B., Malinverno, P., Pezzini, M., Sholler, D., Natis, Y., & Iijima, K. (2010). *Magic Quadrant for Application Infrastructure for Systematic Application Integration Projects*. 18 de octubre de 2010. URL: <http://www.gartner.com/techn>. Recuperado de: <http://www.gartner.com/technology/media-products/reprints/softwareag/volume2/article7/article7.html>

Vollmer, K., Gilpin, M., & Rose, S. (2011). *The Forrester Wave™: Enterprise Service Bus*. Recuperado de: <http://www.slideshare.net/jricardoferreira/the-forrester-wave-enterprise-service-bus-q2-2011>

Wu, J., & Tao, X. (2010). Research of enterprise application integration based-on ESB. *Advanced Computer Control (ICACC), 2010 2nd International Conference on*, 90-93. Recuperado de: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5487292&isnumber=5486785>