

ARQUITECTURA DEL KNOWLEDGE SHARED SYSTEM 2.0

David Cebrián Tarrasón

Rosario Vidal Nadal

GID, Grupo de Ingeniería del Diseño, Dpto. Ingeniería Mecánica y Construcción, Universitat Jaume I de Castellón (España)

Abstract

Currently, at the time of the current crisis, it is more necessary than ever to address a full optimization of resources in all processes within an organization. This is more evident in the stage of collected knowledge by the other staff because they are obtaining a series of routines through their work experience. This means that companies must address both development and knowledge management of designed and manufactured products being this activity one of the highest priority.

Knowledge Sharing System 2.0 (KSS 2.0) is a system of Knowledge Based Engineering (KBE). KSS 2.0 enables the access to formalized information from a product design in real time by the client, as the company and the designer.

KSS 2.0 is adapted for the introduction of information using a web portal that links directly to an ontology using new web technologies available such as RIA (Rich Internet Applications).

This article explains the basis of architecture and design of this tool.

Keywords: *Knowledge Based Engineering (KBE); MOKA Methodology; Ontology Engineering; product design*

Resumen

Actualmente, en la época de crisis actual, es más necesario que nunca abordar una optimización completa de recursos en todos los procesos dentro de una organización. Este hecho se hace más evidente en el apartado del conocimiento recogido por el distinto personal que a través de su experiencia laboral adquiere una serie de rutinas de trabajo. Esto significa que las empresas deben abordar tanto el desarrollo como la gestión del conocimiento de los productos que diseñan y fabrican, siendo esta actividad una de las más prioritarias y fundamentales.

El sistema Knowledge Sharing System 2.0 (KSS 2.0) es un sistema de Ingeniería Basada en el Conocimiento (KBE -Knowledge Based Engineering) que posibilita el acceso a la información formalizada del diseño de un producto en tiempo real tanto por parte del cliente, como de la empresa y el diseñador.

Está adaptada para la introducción de información utilizando un portal web que se enlaza directamente con una ontología haciendo uso de las nuevas tecnologías web disponibles tales como RIA (Rich Internet Applications - Aplicaciones de Internet Enriquecidas).

En este artículo se explica la arquitectura y bases de diseño de dicha herramienta.

Palabras clave: *Ingeniería Basada en el Conocimiento (KBE); Metodología MOKA; Ingeniería Ontológica; Diseño de Productos Industriales*

1. Introducción

La ingeniería basada en el conocimiento (KBE) es una disciplina con base en CAx (Computer Aided Technologies– Tecnologías Basadas en el Ordenador) y sistemas basados en el conocimiento (KBS- Knowledge-Based Systems) pero que tiene diversas definiciones y funciones dependiendo del contexto de su aplicación (Penoyer et al., 2000). En el ámbito de la ingeniería del diseño, KBE se puede definir como el desarrollo de una estructura, a partir de la cual se puede implementar un diseño automático, haciendo uso del conocimiento experto sobre el ciclo de vida de un producto (Vidal and Mulet, 2006).

Por ello, un sistema KBE requiere almacenar el conocimiento necesario para el posterior desarrollo de una aplicación para la cual se ha creado. Igualmente, en este ámbito se han desarrollado trabajos entorno a la aplicación de este tipo de reglas aplicadas al ámbito médico (Cebrián-Tarrasón et al., 2007).

A pesar de los estudios llevados a cabo para la aplicación formal del conocimiento en el proceso de diseño, se han creado pocas soluciones complejas de sistemas KBE que permitan integrar la aplicación del conocimiento con las funciones modernas de los sistemas CAx. Este tipo de aplicaciones se considerarían sistemas abiertos, independientes de cualquier otra aplicación (Vidal and Mulet, 2006, Skarka, 2007). Por otro lado, hay un grupo de sistemas CAx que tienen la posibilidad real de representar y almacenar el conocimiento en forma de funciones listas para usar. La principal base para su aplicación es la conexión perfectamente establecida entre el conocimiento obtenido de procesos particulares a través de reglas, entidades estructurales y funcionales del producto de diseño y sus restricciones.

Una ontología se puede describir como una especificación explícita de una conceptualización compartida, la cual se puede basar en una taxonomía o en axiomas (Gruber, 1993). Las ontologías se pueden basar en una sola taxonomía o en varias siendo conceptos y relaciones que se organizan jerárquicamente y cuyos conceptos pueden ser ordenados como clases e instancias (Gómez-Pérez, 2004). La estructura de una ontología debe estar basada en una taxonomía que tenga en cuenta el modelado de un sistema basado en ciertas descripciones funcionales (Garbacz, 2006). De este modo, se han modelado una gran diversidad de metodologías para el análisis del proceso de diseño basadas en el marco FBS (Cebrián-Tarrasón, 2008a).

En este artículo se presenta la arquitectura y las bases de diseño del trabajo presentado en Cebrián-Tarrasón (2009) para demostrar la estructura de funcionamiento del Knowledge Sharing System (KSS) 2.0 posibilitando el acceso a la información en tiempo real tanto por parte del cliente, como de la empresa y el diseñador de toda la información. Para ello inicialmente se recuerda ciertos antecedentes del KSS 2.0, para situar al lector. Posteriormente se indica la arquitectura del KSS 2.0. Finalmente se explica en profundidad la programación en Java utilizada y se muestran las conclusiones.

2. Antecedentes del KSS 2.0

Para poder comprender el funcionamiento del KSS 2.0, se recomienda el artículo de Cebrian-Tarrason (2010) donde se introducen una serie de nociones de MOKA, OntoFaBES y Web 2.0.

3. KSS 2.0

Con el objetivo de integrar la captura del conocimiento de la empresa extendida en un entorno web, se ha desarrollado KSS 2.0 (Knowledge Sharing System 2.0), (Cebrián-Tarrasón 09).

Este sistema permite constituir una plataforma en la que toda la información sobre el ciclo de vida del producto está disponible tal como se puede observar en la Figura 1. Tal como se indica:

- El sistema KSS 2.0, el cual persiste toda la información disponible, se logra una comunicación directa con clientes y proveedores;
- La macro SW-KSS, una macro desarrollada para el programa de CAD (Computer Aided Design – Diseño Asistido por Ordenador) Solidworks, posibilita al diseñador disponer de la información del ciclo de vida del producto;
- La ontología OntoFaBES, permite consultar todo esta información por el ingeniero del conocimiento responsable del mantenimiento de la plataforma.

A través de este sistema disponible sobre un navegador web se consigue una mejor especificación de requisitos y utilidades de los productos en diseño y de los productos adquiridos a proveedores, aumentado por ello el valor añadido del producto final.

La librería de modelos para este sistema está basada en el sistema OntoFaBES, que engloba la ontología principal del sistema, y el modelo KSS 2.0. Este sistema permite gestionar el conocimiento proveniente del software de CAD Solidworks, y su mantenimiento a través del software Protégé, editor de ontologías.

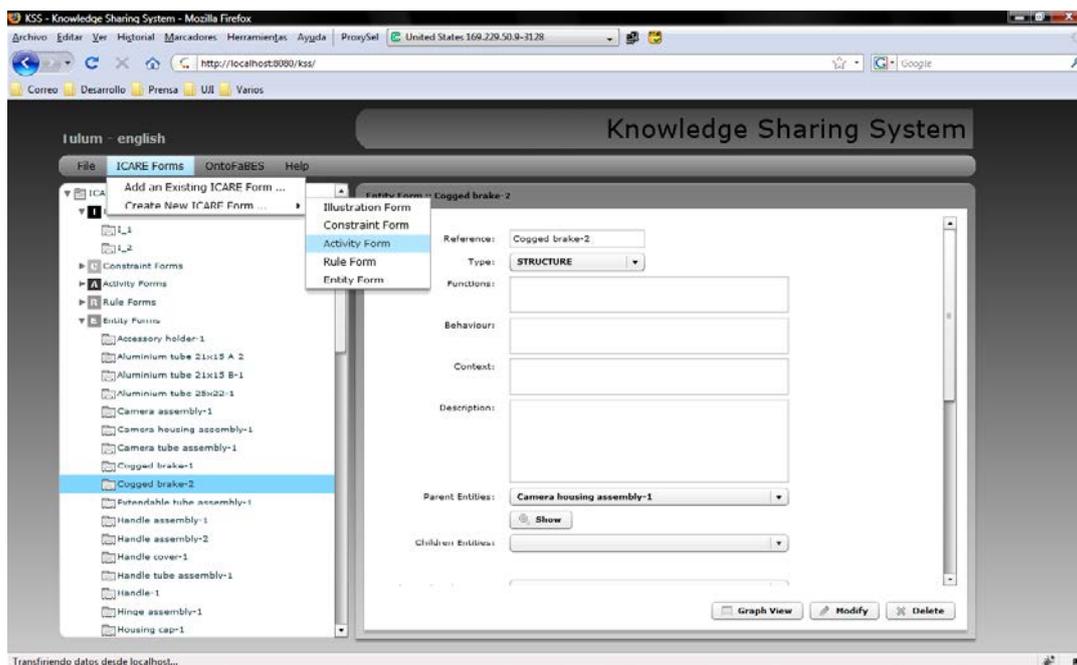


Figura 1: KSS 2.0

Un ejemplo práctico de ello describió Cebrián-Tarrasón (2010) respecto a la aplicación del KSS 2.0 a una sección modular de una cámara submarina, donde se determina el acceso a la información formalizada en tiempo real tanto por parte del cliente, como de la empresa y el diseñador. Dicho flujo de información permite una posible reducción de costes en procesos de rediseño industrial, y mayor flexibilidad a la hora de realizar posibles cambios, al adaptarse a las necesidades de todas las partes interesadas.

4 ARQUITECTURA DEL KSS 2.0

La arquitectura del KSS 2.0 está dividida en dos partes claramente diferenciadas: una parte cliente y otra parte servidora.

La parte cliente la forman los distintos clientes que se benefician de las funcionalidades de la arquitectura. OntoFaBES proporciona dos clientes:

- KSS Web Client
- SolidWorks Client

El cliente KSS Web Client es el cliente nativo del sistema KSS. Los usuarios podrán consultar proyectos de diseño y gestionar los formularios ICARE de los mismos desde esta aplicación web. Esta aplicación web es el front-end genérico de KSS, y está desarrollada utilizando tecnologías Web 2.0 como es la RIA (Rich Internet Application) Flex de Adobe.

El cliente SolidWorks Client es un cliente auxiliar del sistema KSS. Su principal función es la conversión y persistencia en el sistema, de forma rápida y ágil, de formularios ICARE. En este caso, los usuarios de SolidWorks cargarán un determinado proyecto de diseño y de forma semi-asistida se persistirán los formularios ICARE asociados a cada estructura del proyecto. Este cliente se trata de una macro de la herramienta de CAD SolidWorks, desarrollada en Visual Basic for Applications (VBa).

La parte servidora está compuesta por el núcleo de la aplicación, la funcionalidad y lógica de negocio del sistema. Podemos distinguir dos capas dentro de esta parte de la arquitectura:

- Interfaz Web Service
- Back end

La interfaz Web Service de la parte servidora posibilita la interoperabilidad y la integración entre distintos clientes, permitiendo de esta manera poder trabajar con distintas herramientas o sistemas CAD. La comunicación cliente – web service se logra utilizando el protocolo SOAP, protocolo empleado en los servicios web. La interfaz web service está desarrollada en Java, utilizando el API JAX-WS.

La capa de Back end implementa toda la lógica de negocio del sistema, como es la gestión de los formularios ICARE y la de la ontología OntoFaBES. Para la gestión de la ontología OntoFaBES se optado por utilizar el API de Protégé 3.4, junto al razonador JESS de esta forma tener soporte para ejecutar consultas SPARQL y ejecutar reglas SWRL.

El desarrollo de esta capa también ha sido realizado en Java, en este caso implementado bajo la plataforma Java EE. La capa de back end incorpora también una capa de persistencia, que se ha desarrollado utilizando el frameworkORMHibernate, utilizando como base de datos MySQLRDBMS 6.0.

El cliente KSS Web Client y la parte servidora van desplegadas en un servidor de aplicaciones, en este caso JBOSS Application Server debido a su buena operatividad con la tecnología Hibernate, apoyo comunitario y una elevada cuota de mercado (Roth, 2006).

El esquema y estructura del sistema KSS queda reflejado en la Figura 2.

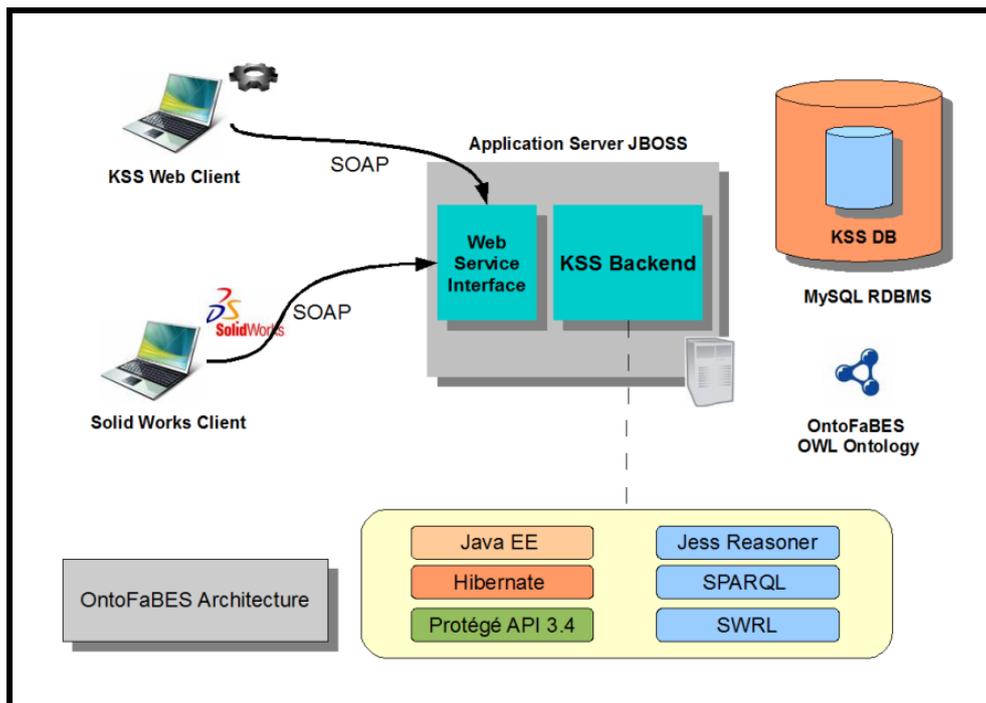


Figura 2: Implementación de la arquitectura OntoFaBES

4.1 Macro KSS-Solidworks

La macro de KSS hace uso del lenguaje programación VBA de Solidworks. Dicho lenguaje utiliza una API para lograr el reconocimiento de las partes del modelo del diseño creado. Para ello se basa en las características del modelo en la forma de 3D. La Figura 3 indica el procedimiento relativo a la integración de la preparación de la parte sólida, el reconocimiento de sus piezas y las conexiones establecidas entre ellas.

Con la información disponible por parte de Solidworks a través de la macro KSS (Figura 4) se generan dos apartados diferenciados:

- Formulario de la interfaz del usuario: A través de ese formulario se muestra la información obtenida (Figura 1).
- Módulos de gestión de la información: Permiten la disposición de la información relevante para el sistema KSS 2.0, la cual se estructura en 4 partes:
 - Módulo de conexión: Estructura los tipos de conexión y alineamiento que se pueden realizar entre las distintas piezas.
 - Tabla de restricciones: Indica las diferentes restricciones existentes cuando se unen dos o más piezas.
 - Árbol de entidades: Estructura la jerarquía de las piezas del diseño de un objeto en forma de árbol.
 - Permiso de conexiones: Establece una serie de reglas que permiten las conexiones adecuadas entre piezas.



Figura 3: Procedimiento de reconocimiento de las partes de Solidworks

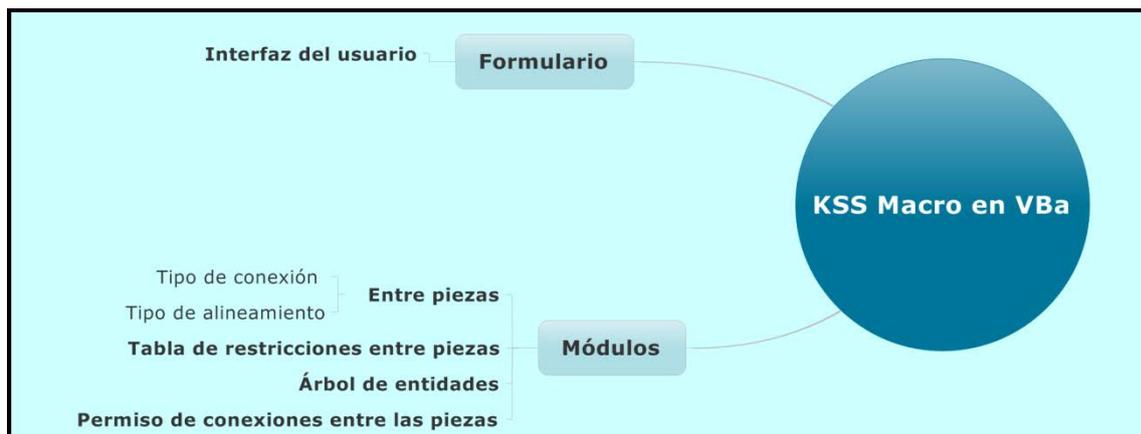


Figura 4: Esquema de la macro KSS

4.2 Programación en Java – KSS Java Client

KSS 2.0 a su nivel de programación en java está desarrollado en los siguientes módulos (Figura 5):

- **Data:** Este módulo se encarga de gestionar los datos necesarios para el adecuado funcionamiento del programa.
- **Back-endPersistence:** Se encarga de persistir la información del proyecto.
- **Util:** Se encarga de organizar la información según la estructura de MOKA y la ontología OntoFaBES.
- **BI (Business Intelligence):** Gestiona la interfaz e implementación del proyecto y de los formularios MOKA.
- **Web Service:** Implementa el servicio web de la aplicación.

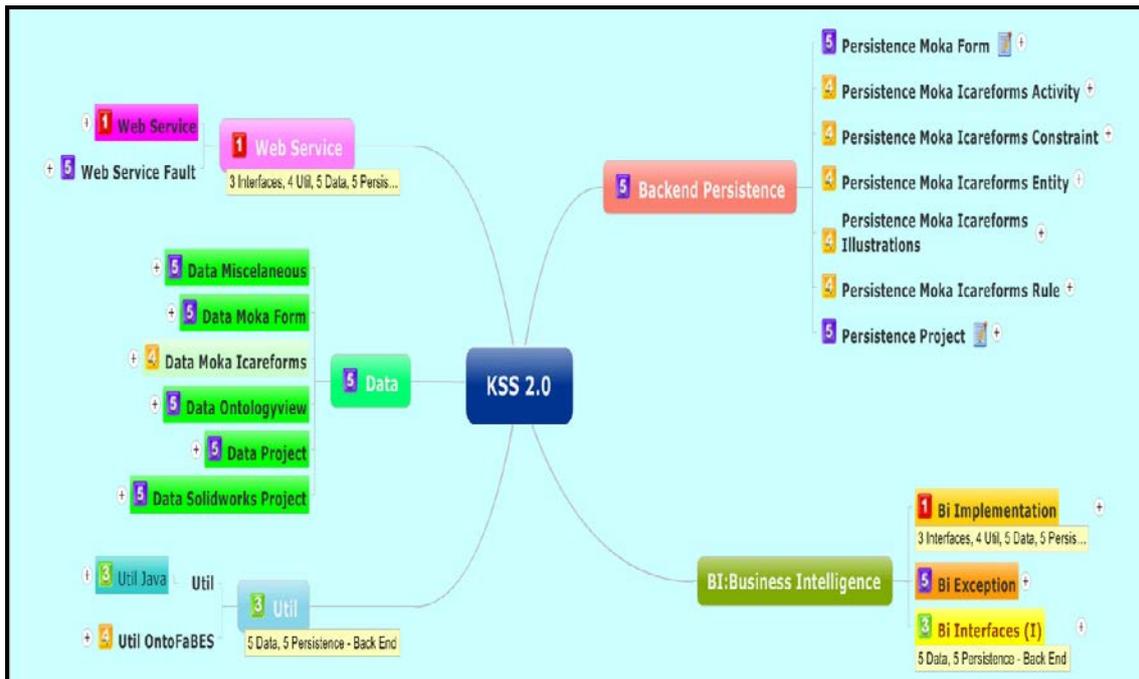


Figura 5: Organización del sistema KSS 2.0

Los distintos módulos están interrelacionados con diferentes dependencias debido a las necesidades de información del sistema. Tal como se observa en la Figura 6 a partir del módulo de datos y el de la persistencia de la información se estructuran las conexiones del sistema.

A continuación se explicará en profundidad las características de cada uno de los módulos.

4.2.1 Capa de Data (Java)

La capa de datos (Data) está conformada por 6 paquetes distintos:

- **Data Miscellaneous:** Este paquete se encarga de implementar las funciones necesarias para visualizar los grafos de los formularios ICARE (Cebrian-Tarrason 2010) en la aplicación web.
- **D. Moka Form:** Gestiona la implementación de la plantilla del formulario necesario para estructurar la información de los distintos formularios ICARE.
- **D. Moka Icareforms:** Gestiona los datos de los formularios ICARE.
- **D. Ontologyview:** Organiza la información sobre la visualización de la ontología en el sistema.
- **D. Project:** Gestiona el proyecto referido a los datos del sistema KSS 2.0.
- **D. Solidworks Project:** Gestiona el proyecto referido a los datos provenientes de Solidworks referidos a la macro KSS.

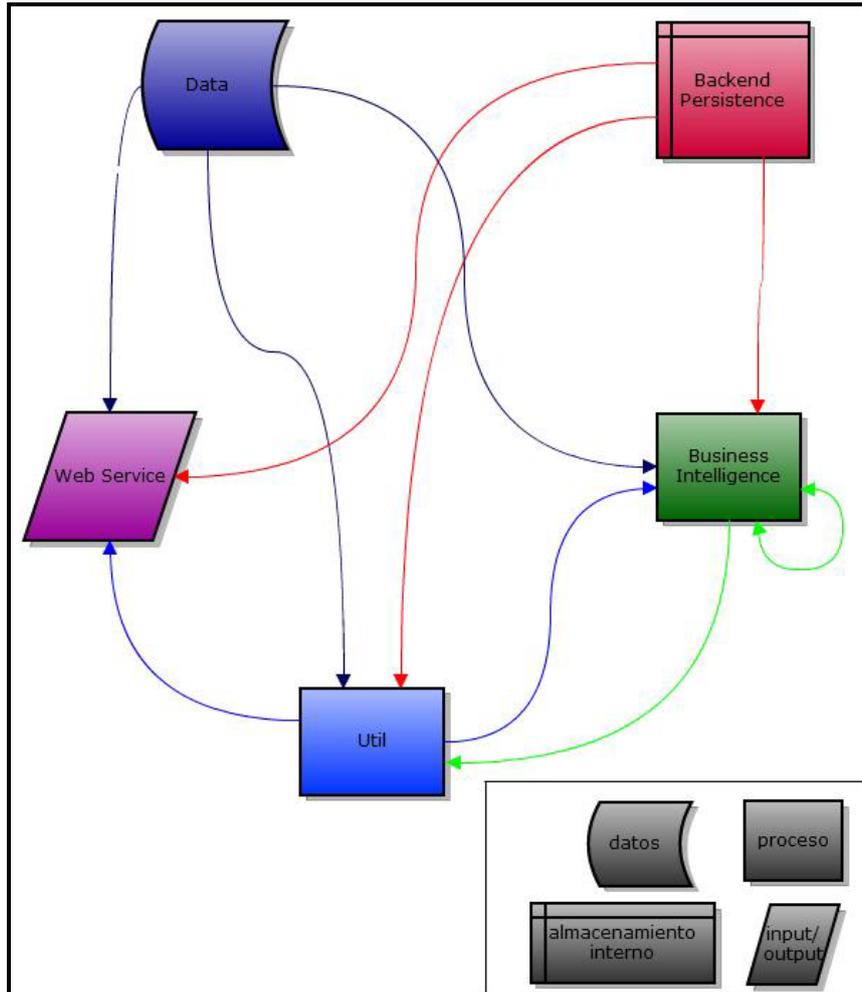


Figura 6: Conexiones del sistema KSS 2.0

4.2.2 Capa de Back End– Persistencia (Java)

La capa de Back End está conformada por 7 paquetes distintos:

- **Persistence Moka Form:** Gestiona la plantilla del formulario necesario para estructurar la persistencia de información de los distintos formularios ICARE.
- **P. M. IcareformsActivity:** Gestiona la persistencia de datos del formulario Activity.
- **P. M. I. Constraint:** Gestiona la persistencia de datos del formulario Constraint.
- **P. M. I. Entity:** Gestiona la persistencia de datos del formulario Entity.
- **P. M. I. Illustrations:** Gestiona la persistencia de datos del formulario Illustrations.
- **P. M. I. Rule:** Gestiona la persistencia de datos del formulario Rule.
- **Persistence Project:** Gestiona el proyecto referido a la persistencia de datos del sistema.

4.2.3 Parte Útil

La capa de la parte Util está conformada por 2 paquetes distintos:

- **Util:** Establece las funciones necesarias para la distribución de la información entre los formularios ICARE en la aplicación web.
- **Util OntoFaBES:** Permite la ejecución de la ontología OntoFaBES en el sistema KSS 2.0. Por tanto sus dependencias están basadas en la API Protégé y en la información proveniente de Solidworks.

4.2.4 Parte Lógica de Negocio

La capa de la lógica de negocio está conformada por 10 paquetes distintos organizados en 3 grupos:

- **Bi Implementation:** Este grupo de 3 paquetes se encargan de implementar la lógica de negocio del KSS 2.0. Estos paquetes son los siguientes:
 - **Bi Implementation Moka Form:** Gestiona la implementación de la plantilla del formulario necesario para los distintos formulario ICARE.
 - **B. I. Moka Icareforms:** Gestiona la implementación de los formularios ICARE.
 - **B. I. Project:** Es el paquete que gestiona el proyecto de implementación.
- **Bi Exception:** Crea una función para el adecuado tratamiento de las excepciones .
- **Bi Interfaces:** Está conformado por un grupo de 3 paquetes que se encargan de gestión de los formularios en la interfaz de KSS 2.0.

El paquete de implementación de los formularios ICARE (B. I. Moka Icareforms) y la referida a la interfaz (Bi Interfaces), se explican en profundidad a continuación. Así, la parte de Implementación está conformada por 5 clases distintas dedicadas a cada uno de los formularios ICARE y la parte de interfaz está conformada por 3 paquetes distintos:

- **Bi Interfaces Moka Form:** Gestiona la plantilla de la interfaz necesaria para los distintos formularios ICARE.
- **Bi Interfaces Moka Icareforms:** Gestiona la interfaz de los formularios ICARE.
- **Bi Interfaces Project:** Es el paquete que gestiona el proyecto de la interfaz.

4.2.5 Web Service

La capa de Web Service está conformada por 2 paquetes distintos: Web Service y Web Service fault que constituyen el apartado del servidor web del KSS 2.0.

5. Conclusiones

Este artículo muestra la arquitectura y base de diseño de KSS 2.0 centrándose en la lógica de programación.

Este trabajo quiere mostrar la implementación de la arquitectura de una ontología en un sistema más complejo en el que se entrelazan tres tipos diversos de tecnologías: los KBE al obtener información de un software CAE, las ya citadas ontologías y los entornos RIA como ejemplo de la evolución de la Web 2.0 en el mundo de Internet. Se ratifica con ello que es posible formalizar la información de un diseño con tal sólo poder tenerlo disponible en un software CAE (Ostad-Ahmad-Ghorabi, 09). Al formalizar esta información se hace extensible

para ser utilizada por una ontología, en este caso, OntoFaBES y visualizable en Internet con las ventajas de coste que eso permite.

En este trabajo también se ha demostrado que la arquitectura permite que OntoFaBES sea aplicable en diversos diseños.

En resumen, este trabajo permite afirmar que la web 2.0 llega al ámbito de la ingeniería del diseño a través de las virtudes de la compartición de información.

6. Referencias

Cebrián-Tarrasón D. and Vidal R. (2010). *Muestra del Knowledge Shared System 2.0: Aplicación a una pieza mecánica*. XIV Congreso Internacional de Ingeniería de Proyectos (AEIPRO).

Cebrián-Tarrasón D. and Vidal R. (2009). *KSS 2.0: Knowledge Shared System aplicado a la Web 2.0*. XIII Congreso Internacional de Ingeniería de Proyectos (AEIPRO).

Cebrián-Tarrasón, D., Vidal, R. (2008a), *Las ontologías y el marco FBS*, XII Congreso Internacional de Ingeniería de Proyectos (AEIPRO).

Cebrián-Tarrasón D. and Vidal R. (2008b). *How an ontology can infer knowledge to be used in product conceptual design*. Computer-Aided Innovation - IFIP International Federation for Information Processing, 57-68. doi: 10.1007/978-0-387-09697-1.

Cebrián-Tarrasón D., Muñoz C., Chulvi V. and Vidal R. (2007), *Nuevo enfoque en el diseño inteligente de implantes craneales personalizados a través de KBE*. XI Congreso Internacional de Ingeniería de Proyectos (AEIPRO), 668-677.

Chui, M., Miller, A., Roberts, R.P. (2009): *Six Ways to make Web 2.0 work*. The McKinsey Quarterly.

Garbacz P. (2006), *Towards a Standard Taxonomy of Artifact Functions*, Applied Ontology, Vol.1 (3), 221-236.

Gómez-Pérez A., Fernández-López M. and Corcho O. (2004), *Theoretical Foundations of Ontologies - Chapter 1*, Springer, ed., Ontological Engineering with Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web, London, 1- 44.

Gruber T.R. (1993), *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*, Knowledge Systems Laboratory, Stanford University., Padova, Italy.

Oreilly, T. (2007), *What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. Communications & Strategies, No. 1, p. 17.

Ostad-Ahmad-Ghorabi, H., Collado-Ruiz, D., Wimmer, W. (2009) *Towards Integrating LCA into CAD*. In: Proceedings of the 17th International Conference on Engineering Design, Norell, M., Grimheden, M., Skogstad, P. (editors), p: 301-310, ISBN 978-190-4670-11-7.

Pc Magazine, (2007), *La Nueva Red*, Agosto 2007.

Penoyer, J., Burnett, G., Fawcett, D. and Liou, S. (2000), Knowledge based product life cycle systems: Principles of integration of KBE and C3P, *Computer-Aided Design*, 32, 311-320.

Skarka, W. (2007). Application of MOKA methodology in generative model creation using CATIA, *Engineering Applications of Artificial Intelligence*, 20:5, 677-690.

Stokes, M. (2001). *Managing Engineering Knowledge: MOKA Methodology and Tools for Knowledge based Engineering Application*. Cambridge University.

Vidal, R. and Mulet, E. (2006), Thinking about computer systems to support design synthesis, *Communications of the ACM*, 49, 100-104.

Wikipedia, 2010. http://en.wikipedia.org/wiki/Web_2.0#Criticism Obtenido en 6/04/2010.

Correspondencia (Para más información contacte con):

GID (Grupo de Ingeniería del Diseño) Departamento de Ingeniería Mecánica y Construcción
Universitat Jaume I
Phone: 627573200
E-mail: dazebri@gmail.com
URL: <http://www.gid.uji.es>