

Hacia la implementación de una herramienta de soporte al proceso de desarrollo de software

Larburu¹ I.U., Pikatza¹ J.M., Sobrado¹ F.J., García² J.J. y López de Ipiña³ D.

¹Departamento de Lenguajes y Sistemas Informáticos (UPV/EHU)^a
{jiblaeni, jippiacj, jibsootf} @ sc.ehu.es

²Wireless Systems Research Department, Avaya Labs Research, Basking Ridge, NJ (USA)
jjga@avaya.com

³Consulting Department, 3G LABS, Cambridge (England)
dipina@3glab.com

Abstract. La implantación de procesos desarrollo del software (PDS) en una organización aumenta su rendimiento y capacidad de negocio. La disposición de un Sistema de Ayuda a la Toma de Decisiones (DSS) vía Web que implemente el proceso resulta una solución para la implantación de la disciplina exigida. Además de definir un PDS para la fase de elaboración de proyectos, presentamos una forma de representación y ayuda a la decisión en el PDS en base al formalismo Guidelines Interchange Format (GLIF) y al método de asignación difusa multicriterio PROAFTN. La ontología del dominio del PDS (SPOnt) integra las de GLIF, MCDA y PROAFTN. De esta manera, además de aprovechar la capacidad de definición de guías de GLIF y la ayuda a la decisión de PROAFTN, define relaciones entre tareas, roles y artefactos. Proponemos una arquitectura de DSS que integra los elementos anteriores con módulos para la gestión de datos de evolución.

1 Introducción.

La disposición de un proceso de desarrollo del software (PDS) definido e implantado en una organización aumenta su rendimiento y capacidad de negocio. La implementación del proceso en un Sistema de Ayuda a la Toma de Decisiones (DSS) accesible vía Web puede ser una estrategia de implantación efectiva de la disciplina subyacente si, además, ayuda en la gestión de los datos de evolución del proceso. Existen unas 150 empresas en el mundo en los niveles 4 o 5 de SW-CMM, que han dado una solución a este problema. No hemos detectado DSSs de soporte al PDS pero sí aportaciones parciales en la línea de la gestión de experiencia adquirida [3], gestión de proyectos^b, soporte al trabajo en grupo [11], evaluación de procesos [4], etc.

Para la implementación de un proceso, se recomiendan soluciones flexibles como los *workflows* adaptativos [2] usando meta-modelos como el propuesto por Papavassiliou [6] que integra tareas de conocimiento y de desarrollo. Existen

^a Fac. Informática (UPV/EHU), Dep. LSI, Apdo. 649, 20080 Donostia – San Sebastián

^b Websystems Inc. <http://www.project-management-software.org>

múltiples formalismos de representación utilizables: gramáticas formales, redes de Petri, y los diagramas de IDEF^c, BPML^d o el *Guidelines Interchange Format* (GLIF) [7].

Dado el protagonismo alcanzado por los métodos de clasificación híbridos basados en la lógica difusa englobados en lo que se conoce como Ayuda a la Decisión Multicriterio (MCDA) y *Soft-Computing*, utilizados por nosotros para la implantación de guías de práctica clínica en Telemedicina con el DSS Arnasa [10], recurrimos también a ellos para la construcción del DSS de soporte al PDS.

En la siguiente sección detallamos los materiales y métodos que hemos utilizado en nuestra investigación, en la tercera sección hacemos hincapié en los resultados obtenidos y terminamos con una cuarta sección en la que apuntamos las conclusiones obtenidas.

2 Materiales y métodos.

Dentro de los trabajos de construcción del módulo de ayuda a la decisión en el PDS, hemos definido un proceso para la fase de elaboración de proyectos aplicable en organizaciones de pequeño tamaño, adaptable al crecimiento de los mismos. El dominio de la fase de elaboración de un proyecto (Fig. 3). Este proceso, tiene en cuenta conceptos de otras metodologías (RUP [5] y CommonKADS [8]) y pretende cumplir con los requisitos de nivel 2 de los modelos SW-CMM^e y CMMI^c.

Para obtener una representación flexible de procesos de conocimiento intensivo débilmente estructurados extendemos el meta-modelo de Papavassiliou [6] con las clases *Artifact*, *Knowledge role* y *Development role*. La ontología resultante la integramos con la ontología de GLIF [7] por su utilidad en nuestra línea de telemedicina, su posibilidad de integrar vocabularios, y su estructuración en tres niveles: conceptual, computable, y de integración con el entorno.

Para ofrecer ayuda a la decisión hemos optado por un método de Ayuda a la Decisión Multicriterio (MCDA) denominado PROAFTN por el poder clasificatorio demostrado en tareas de diagnóstico médico, donde se muestra como el mejor clasificador entre técnicas como árboles de decisión, reglas de producción, K-NN, regresión logística y perceptrones multicapa según la comparativa de Belacel [1].

PROAFTN, no calcula repetidamente las distancias y admite datos en diferentes unidades. Al basarse en principios de concordancia y no discordancia, se engloba dentro de los algoritmos no totalmente compensados, utilizando aprendizaje tanto deductivo como inductivo, y criterios tanto cuantitativos como cualitativos.

PROAFTN define un conjunto $\mathcal{C} = \{C^1, C^2, \dots, C^k\}$ de clases, que tienen asociado un conjunto de prototipos $B^h = \{b_1^h, b_2^h, \dots, b_{L^h}^h\}$ que se definen como tuplas sobre un conjunto $F = \{g_1, g_2, \dots, g_n\}$ de atributos, y una matriz W , donde W_j^h representa el peso que tiene el atributo g_j en la categoría C^h . Todos los pesos son valores positivos en el rango $[0,1]$, y la suma de todos los pesos de una categoría es 1.

^c IDEF Family of Methods. <http://www.idef.com>

^d Business Process Management Initiative. <http://www.bpmi.org>

^e Software Engineering Institute. <http://www.sei.cmu.edu/cmm/cmms/cmms.html>

Las funciones $C_j(a, b_i^h)$, concordancia del atributo g_j entre el elemento a y el prototipo b_i^h , y $D_j(a, b_i^h)$ discordancia (Fig. 1), se basan en 3 pares de valores: $S_j^-(b_i^h)$ y $S_j^+(b_i^h)$ forman el rango de máxima concordancia entre el atributo g_j y b_i^h , los umbrales de discriminación $d_j^-(b_i^h)$ y $d_j^+(b_i^h)$ ante la posible imprecisión de los datos, y los umbrales veto $v_j^-(b_i^h)$ y $v_j^+(b_i^h)$ para señalar la zona de incompatibilidad.

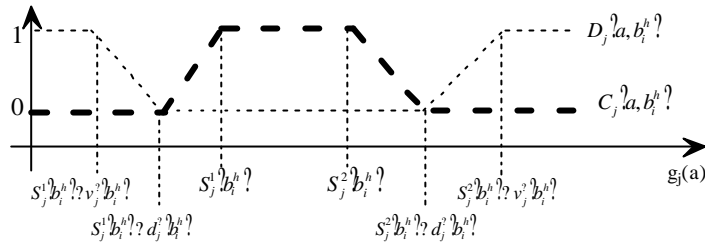


Fig. 1. Representación gráfica de C_j y D_j

El grado de semejanza entre el elemento a y la clase C^h es el mayor de los grados de semejanza entre el elemento y los prototipos de la clase b_i^h (1).

$$I(a, b_i^h) = \max_{j=1}^n C_j(a, b_i^h) \quad (1)$$

Para la implementación del prototipo de DSS para la gestión del PDS llamado SPK hemos utilizado la especificación J2EE de la tecnología Java, Protégé-2000 v1.7^f como entorno de desarrollo de ontologías y herramienta de adquisición de conocimiento por el buen nivel de funcionamiento demostrado en múltiples aplicaciones [8], JESS v6.1^g como motor de inferencia por su amplio uso, incluyendo la extensión para lógica difusa FuzzyJESS^h, y por JessTab^g como puente de comunicación entre Protégé-2000 y JESS.

3 Resultados.

El prototipo de SPK construido, presenta la arquitectura de la Fig. 2. Al igual que en el DSS Arnasa [10], su funcionalidad esta repartida entre diversos módulos Web especializados que, además de la ayuda a la decisión, cubren aspectos como la visualización interactiva 2D y 3D de datos de evolución del proceso, análisis de datos, control de acceso basado en roles, gestión de planes, comunicación entre roles para el trabajo en grupo, o la internacionalización de la interfaz.

Utilizamos Protege-2000 tanto a nivel de administrador como a nivel de usuario del DSS. El administrador edita las ontologías mediante Protege-2000, el usuario define sus propios procesos creando instancias de las clases definidas.

La información contenida en las ontologías es convertida en una estructura simple aceptable por el motor de inferencia JESS mediante el *plug-in* JessTab de Protege-

^f Protégé-2000. <http://protege.stanford.edu>

^g Jess, the rule engine for Java™ Platform. <http://herzberg.ca.sandia.gov/jess/>

^h NRC-IIT. http://www.iit.nrc.ca/IR_public/fuzzy/fuzzyJToolkit2.html

2000. JESS es el responsable de la ejecución tanto del proceso definido sobre la ontología, como de cualquier elemento de ayuda a la decisión que el DSS pueda ofrecer.

Para la implementación del proceso hemos creado una ontología (SPOnt) basada en el meta-modelo de *workflow* de Papavassiliou [6]. A este modelo le hemos añadido el concepto de artefacto (*Artifact*) y hemos dividido el concepto rol en dos: los roles de conocimiento y los roles de desarrollo (*Knowledge role* y *Development role*).

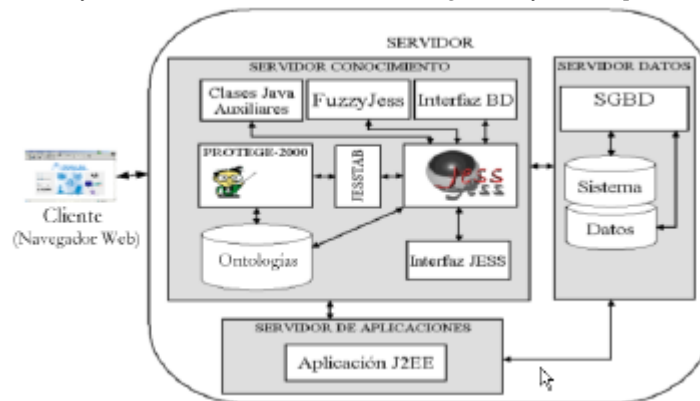


Fig. 2. Arquitectura del DSS para la gestión de procesos SPK.

GLIF define guías (*Guidelines*) que se ejecutan siguiendo algoritmos (*Algorithm*), que se componen de pasos (*Guideline steps*) que se pueden descomponer en subguías. El núcleo de SPOnT (Fig.3) está definido sobre 4 conceptos: proceso, tarea, rol y artefacto. Hemos definido *proceso* como una guía y *tarea* como un *paso*. Los roles y los artefactos son conceptos propios de SPOnT y no tienen reflejo en el ámbito de GLIF. Las *tareas*, realizadas por un *rol* concreto, utilizan y producen *artefactos*.

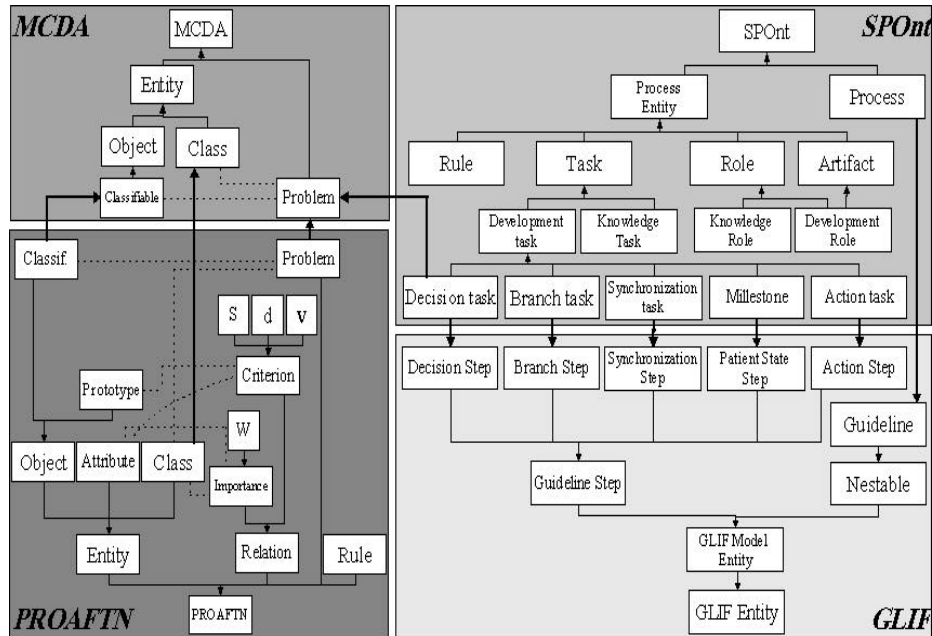


Fig. 3. Relación entre las ontologías SPont, GLIF, MCDA, PROAFTN

La Fig. 4 muestra la relación entre las disciplinas del proceso. El núcleo de proceso esta compuesto por las disciplinas: *Gestión de Requisitos* – RM (SW-CMM, CMMI y RUP), *Análisis y Diseño* – AD (RUP), *Planificación de Proyecto* – PP (SW-CMM, CMMI y RUP). Como disciplinas auxiliares hemos definido las siguientes: *Gestión de Configuración* – CM (SW-CMM, CMMI y RUP), *Seguimiento y Supervisión de Proyectos* – PTO (SW-CMM y CMMI), *Aseguramiento de Calidad* – QA (SW-CMM y CMMI), *Análisis de Métricas*– MA (CMMI).

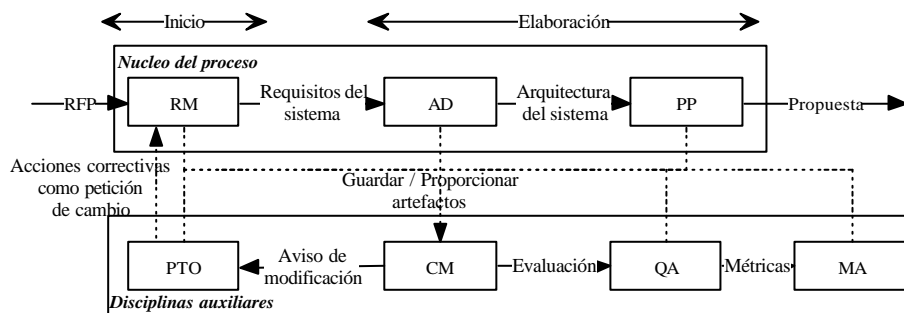


Fig. 4. Relación entre disciplinas del proceso

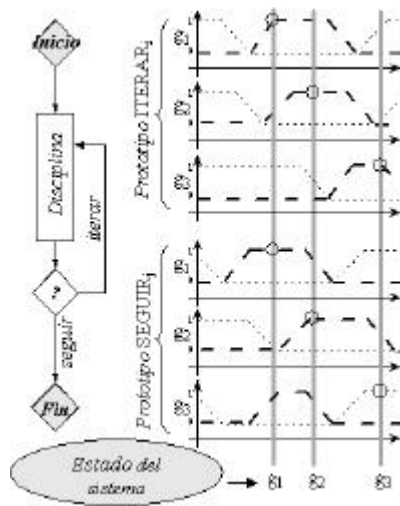


Fig. 5. Ejemplo de ayuda a la decisión

Los procesos definidos sobre SPOnT se parecen al grafo que se muestra en la Fig. 5. Cualquier tarea (nodo) de estos procesos puede a su vez estar descompuesta en diferentes subprocesos.

Los nodos decisión, al tener que resolver un problema (*Problem*) de clasificación, llevan implícita la ejecución de un proceso que implementa un método MCDA; en éste caso, PROAFTN. Como puede verse en la Fig. 4, las clases PROAFTN son las opciones de pasar a otra disciplina (*seguir*) o repetirla (*iterar*). Para cada una, definimos prototipos sobre varios atributos del estado del sistema. El *clasificable* es el estado del sistema en el momento de decidir. En este caso el prototipo *seguir* es vetado por g_3 , por lo tanto el estado actual quedaría asignado a la clase *iterar*.

4 Conclusiones.

La implantación de un PDS permite disponer de información completa de proyectos anteriores y aumenta la predictibilidad, capacidad de negocio, y calidad del proceso y producto final. Sin embargo, el esfuerzo a realizar por los implicados y la falta de herramientas que den una solución global a la gestión de datos, conocimiento y ayuda a la decisión hacen que las organizaciones con PDS implantados sean minoritarias.

El prototipo de DSS construido (SPK), reutilizando varios módulos Web del DSS Arnasa, posibilita un acceso controlado y seguro, internacionalización de la interfaz, visualización de datos de evolución, análisis estadístico de datos, y gestión de datos.

En cuanto a la gestión del conocimiento y ayuda a la decisión, SPK permite tanto el modelado como la ejecución de PDS definidos en base a un conjunto de ontologías (SPOnT, GLIF, MCDA y PROAFTN) interrelacionadas con una capacidad descriptiva suficiente como para definir roles, tareas, artefactos y problemas de decisión en forma de instancias de las clases definidas. El conjunto de tecnologías que integran la solución ofrecida han demostrado una efectividad adecuada a las exigencias de velocidad de respuesta que un DSS accesible vía Web impone.

Este prototipo, de momento y hasta ser utilizado en varios proyectos, es una herramienta de uso interno del Grupo. De cara a su comercialización, es conveniente que las organizaciones usuarias sigan el modelo de desarrollo basado en el dominio. Con ello, además de la ventaja de que los usuarios lleguen a trabajar bajo una disciplina, podrán maximizar los beneficios mencionados anteriormente.

Agradecimientos. Este trabajo ha sido desarrollado con financiación recibida del Departamento de Economía y Turismo de la Diputación Foral de Guipúzcoa [OF-151/2002? bajo el patrocinio de la Unión Europea y, parcialmente, por la recibida del Ministerio de Ciencia y Tecnología [MCYT – TIC2001-1143-C03-01].

Referencias

1. Belacel N. et al: Multicriteria Fuzzy Assignment Method: a Useful Tool to Assist Medical Diagnosis. In Artificial Intelligence in Medicine 21 (2001).
2. Chung P.W.H et al: Knowledge-based process management - an approach to handling adaptive workflow. In Knowledge-Based Systems 16 (2003) 149-160
3. Friedrich R. et al: Experience-based decision support for project management with Case-Based Reasoning. (2002). <http://citeseer.nj.nec.com/527558.html>
4. Henderson P. et al: A tool for evaluation of the software development process. In The Journal of System and Software 59 (2001) 355-362
5. Kruchten P. The Rational Unified Process. An Introduction, 2nd ed., Booch, the Addison-Wesley Object Technology Series, Mass., 2000
6. Papavassiliou G. et al: Integrating Knowledge Modeling in Business Process Management. In proceedings of TAKMA-2002 (2002)
7. Peleg M. et al: Comparing Computer-Interpretable Guideline Models: A Case Study Approach In JAMIA, vol. 10, No. 1,(2002) 52-68
8. Schreiber G. et al: A Case Study in Using Protégé-2000 as a tool for CommonKADS. In Proc. of 12th Int. Conf. on Knowledge Engineering and Knowledge Management (EKAW'2000), (2001)
9. Schreiber G. et al. Knowledge Engineering and Management: the CommonKADS Methodology, Bradford Book, MIT press, 1999
10. Sobrado F.J. et al: Arnasa: una forma de desarrollo basado en el dominio en la construcción de un DSS para la gestión del proceso de tratamiento del asma vía Web. En Actas de las Jornadas de Ingeniería del Software y Bases de Datos (2002).
11. Zaychik V. et al: Capturing communication and Context in the Software Process Lifecycle. In Research and Engineering Design Vol. 14, 2, (2003) 75-88.