

FuL
Especificación de Requerimientos de Software

Alejandro Kondrasky

22 de julio de 2011

Índice

1. Introducción	3
1.1. Propósito	3
1.2. Convenciones del Documento	3
1.3. Audiencia Esperada	3
1.4. Alcance del Producto	3
1.5. Estructura del Documento	3
2. Descripción General	4
2.1. Perspectiva del Producto	4
2.2. Características del Producto	4
2.2.1. Procesamiento continuo de la KB	4
2.2.2. Minimización y optimización de la KB	4
2.2.3. Introducir nuevo conocimiento a la KB	4
2.2.4. Planificar experimentos	4
2.2.5. API para la creación de Plugins	5

1. Introducción

1.1. Propósito

El propósito de este documento es la especificación de requerimientos de software en el marco de la tesis de grado de la carrera Lic. en Cs. de la Computación de FaMAF - UNC denominada *Procesador Lógico para diseño de experimentos de Virología*.

Los requerimientos son provistos por integrantes de FuDePAN en su carácter de autores intelectuales de la solución a implementar y colaboradores de dicha tesis.

1.2. Convenciones del Documento

Las palabras clave *DEBE*, *NO DEBE*, REQUERIDO, DEBERÁ, NO DEBERÁ, DEBERÍA, NO DEBERÍA, RECOMENDADO, PUEDE Y OPCIONAL en este documento son interpretadas como esta descripto en el documento *RFC 2119*.

1.3. Audiencia Esperada

A continuación se enumeran las personas involucradas en el desarrollo de la tesis, los cuales representan la principal audiencia de este documento :

- Dr. Carlos Areces: Director de tesis, FaMAF
- Daniel Gutson: Colaborador de tesis, FuDePAN
- Alejandro Kondrasky: Tesista, FaMAF

1.4. Alcance del Producto

El producto especificado en este documento se denomina *Ful* y su principal objetivo es, dada una *KB* determinada, organizar, analizar, chequear incongruencias en ella y ser capas de con este planificar *experimentos* . A la vez que introducimos nuevos conocimientos a la *KB*, este deberá pasar por los procesos previamente nombrados .

El producto final debe proveer al usuario la capacidad de agregar extensiones capaces de interpretar el conocimiento en la *KB* de maneras apropiadas y de introducir nuevo conocimiento mediante un lenguaje formal definido para ello.

La principal responsabilidad de *FuL* es procesar el conocimiento presente en la *KB*, utilizando las extensiones para interpretarlo, para así obtener una planificación del experimento solicitado. Por otro lado, las extensiones son responsables de interpretar ciertos tipos de conocimiento en la *KB* y decidir si estos son consistentes o no.

En su versión inicial, *FuL* incluye una extensión de *planning* y otra de *DL*.

1.5. Estructura del Documento

La estructura de este documento sigue las recomendaciones de *Guía para la especificación de requerimientos de la IEEE (IEEE Std 830-1998)*. Contiene las siguientes secciones :

- *Sección 2:* Provee una descripción general de los aspectos generales del producto, como la perspectiva de este, características principales.
- *Sección 3:* Describe las interfases del producto, tanto las del usuario como interfases de software.
- *Sección 4:* Organiza y describe los requerimientos funcionales del producto.
- *Sección 5:* Organiza y describe los requerimientos no funcionales.

2. Descripción General

2.1. Perspectiva del Producto

Este producto trata de proveer a la comunidad científica una herramienta para organización, interpretación, verificación y exploración del conocimiento del área de virología que se le otorga , para así poder encontrar incongruencias, conjuntos mínimos que representen el mismo conocimiento y conclusiones derivadas automáticamente de dicho conocimiento.

Otra de sus funciones principales es el planificar experimentos equivalentes a preguntas del tipo $A \implies B$.

2.2. Características del Producto

2.2.1. Procesamiento continuo de la KB

Provee la capacidad de continuar procesando la información presente en la KB para así obtener nuevas conclusiones del conocimiento disponible en ella. Este proceso puede ser detenido y guardado para continuarlo posteriormente.

2.2.2. Minimización y optimización de la KB

Permite reducir el conocimiento repetido y reinterpretar este de manera tal de lograr obtener una nueva KB equivalente minimal.

2.2.3. Introducir nuevo conocimiento a la KB

Permite, mediante un lenguaje formal, introducir conocimiento a la KB el cual sera chequeado para encontrar incongruencias con esta para luego ser aceptado o mostrar la incongruencia encontrada.

2.2.4. Planificar experimentos

Provee la capacidad de planificar un experimento, encontrar las conclusiones a los experimentos intermedios y proponer los experimentos intermedios que se deben realizar para obtener una conclusión para el que fue consultado inicialmente.

2.2.5. API para la creación de Plugins

Provee una API para la creación de plugins, los cuales serán utilizados para interpretar el conocimiento que le es otorgado a dicha base de datos. Esta API permite construir tanto el plugin que interpretara el conocimiento como expandir el lenguaje formal para así poder representar nuevo tipo de conocimiento.