FuL
Especificación de Requerimientos de Software

Alejandro Kondrasky

14 de agosto de 2011

Índice

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 *FuDePAN Logic Processor (FuL)* y su principal objetivo es, dada una *Base de Conocimiento (KB)* del área de virología, organizar, analizar, chequear incongruencias en ella y poder utilizarla para planificar *experimentos* . A la vez que introducimos nuevos conocimientos a la KB, ésta 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. Las extensiones son utilizadas para interpretar ciertos tipos de conocimiento, procesarlo utilizando sus KB internas y devolver nuevo conocimiento a FuL.

En su versión inicial, FuL incluye un *planner* y una extensión de *Lógicas Descriptivas (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 y conclusiones derivadas automáticamente de dicho conocimiento.

Otra de sus funciones principales es la asistencia informática para planificar experimentos equivalentes a preguntas del tipo $A \Longrightarrow B$.

2.2. Características del Producto

Las secciones marcadas con * son opcionales o serán incluidas en trabajos futuros.

2.2.1. Procesamiento de la KB

Provee la capacidad de procesar 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. Introducir nuevo conocimiento a la KB

Permite, mediante el lenguaje formal otorgado por FuL[1], introducir conocimiento a la KB el cual sera chequeado para encontrar incongruencias con esta, para luego ser aceptado o mostrar las encontradas.

2.2.3. 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.

Dichos experimentos deben ser equivalentes a preguntas del tipo $A \Longrightarrow B$, siendo A el punto de partida y B la meta a alcanzar por dicho experimento.

2.2.4. API y SDK para Extensiones

Provee una API y un SDK para la creación de extensiones, los cuales serán utilizados para interpretar el conocimiento que le es otorgado a dicha base de datos.

El API define la forma en la que se intercambiara el conocimiento entre la extensión y FuL. El SDK otorga las librerías necesarias tanto para la construcción de la extensión en si, como también la definición de su lenguaje formal de representación de conocimiento correspondiente.

[1] Dicho lenguaje será traducible a lenguajes de representación de datos.

2.2.5. Optimización de la KB*

Tendrá la capacidad de optimizar la KB. Esta proveerá la eliminación de redundancias y devolverá una versión minimál de la KB original, de la cual se puede deducir todo el conocimiento presente en la original.

2.2.6. KB inicial con conocimientos de virología*

Proveer una KB inicial, la cual contenga conocimientos básicos de virología tales como lo que se encuentran el libros de estudio de nivel básico a intermedio. Restricciones al *scope* de dicha KB están *por ser determinadas (TBD)*[2].

[2]Se propuso restringirlo a los arenavirus y los retrovirus. Se decidirá acorde con las necesidades del cliente y los tiempos disponibles.

3. Interfases

3.1. Interfaces de Sistema

Sera capas de correr al menos en sistemas GNU/Linux, así que por este hecho solo se utilizaran librerías compatibles con sistemas GNU/Linux.

3.2. Interfaces de Usuario

La interfase consiste en una *CLI (Command Line Interface)* y un lenguaje formal, detallados a continuación.

3.2.1. CLI

Este sera tratable como un comando de terminal, al cual se le podrán pasar al menos parámetros para realizar las siguientes operaciones[3]:

1. *Procesar el conocimiento en la KB (V):* Sera la opción por defecto en la cual solo procesara el conocimiento existente en la KB para obtener nuevas conclusiones a los posibles experimentos.
   Dicho proceso podrá ser detenido por el usuario mediante una combinación de teclas preestablecidas por FuL. En cuyo caso guardara el estado actual del procesamiento para ser continuado posteriormente.

2. *Introducir un archivo con conocimiento en la KB (V):* Parsea y chequea consistencia de dicho conocimiento para luego chequear incongruencias de este con la KB. En caso de éxito se agregara a la KB, caso contrario sera rechazado en su totalidad.

3. *Realizar un consulta de un experimento de tipo $A \Longrightarrow B$ (V):* Chequea la valides de dicha consulta. En caso de ser valida FuL responderá con un valor booleano o con una lista de experimentos de tipo $A \Longrightarrow B$ para los cuales no tiene respuesta. En caso de ser invalida mostrara el mensaje de error correspondiente.

4. *Agregar una nueva extensión a FuL (V):* El usuario introducirá el nombre de archivo de la extensión que desea agregar. En caso de que dicho archivo no sea valido o surja algún error sera informado del mismo.

5. *Configurar parámetros relacionados con FuL, su KB y extensiones :* TBD[4]

3.2.2. Lenguaje formal

Es el lenguaje a través del cual el usuario deberá utilizar para introducir conocimiento y realizar consultas a FuL. También es con el cual se presentaran los resultados de las operaciones solicitadas. Dicho lenguaje sera capas de representar conocimiento relacionado con el área de Virología[5] , el cual tendrá la

[3] Las operaciones marcadas con *(V)* tendrán la opción ser puestas en modo verboso, lo que hará que muestren por pantalla mayor información relacionada con su procesamiento.

[4] Dependerá de factores relacionados con el diseño e implementación de la KB, extensiones y otros aspectos relacionados con FuL tales su instalación,etc. Deberá ser resuelto en las etapas anteriormente mencionadas.

[5] Pero no exclusivamente.

propiedad de ser almacenable en una KB y ser interpretado por una extensión DL.

3.3. Interfaces de Hardware

No hay requerimientos especificados.

3.4. Interfaces de Software

El producto tendrá los siguientes módulos :

1. *fulmop :* Este es producto principal, el cual se encarga de interactuar con la KB, fulplanner y las extensiones para realizar las operaciones pedidas por el usuario. Para este propósito fulmop realiza intercambios de mensajes, escritos en el lenguaje formal provisto por el producto, para así obtener nuevas respuestas y continuar el ciclo hasta finalizar la operación solicitada satisfactoriamente[6].

2. *fulplanner :* Es el encargado de organizar el proceso de razonamiento, preguntando a fulmop que consecuencias se puede derivar de un determinado conocimiento. Las respuestas recibidas son agregadas a la planificación del experimento. No es capas de interpretar conocimiento alguno, solo interrelacionarlo en base a las respuestas recibidas de fulmop. Este ultimo sera el encargado de pedirle a fulplanner los experimentos para los cuales requiere respuesta y pedirle el estado actual del grafo de planificación. El lenguaje utilizado por fulplanner es el mismo que utiliza fulmop para representar conocimiento.

3. *fulapi :* Es la API definida para interactuar con las extensiones creadas por terceros. Principalmente permitirá a fulmop realizar preguntas a las extensiones acerca de un determinado conocimiento y que estas le devuelvan una respuesta de la misma forma.

4. *fuldl :* Extensión que permite interpretar conocimiento entendible por lenguajes DL. Actúa de la misma manera que cualquier otra extensión, recibiendo conocimiento a ser interpretado y devolviendo nuevo conocimiento respecto de este.

3.5. Interfaces de Comunicación

No hay requerimientos especificados.

[6] Ver la definición de cada operación en requerimientos funcionales para interpretar que significa ser finalizada satisfactoriamente en cada caso.

Nomenclatura

CLI	Command Line Interface
DL	Description Logics (Lógicas Descriptivas)
Experimento	Pregunta (Mejorar luego)
FuL	FuDePAN Logic Processor
KB	Knowledge Base (Base de Conocimiento)
planner	COMPLETAR
scope	Alcance de cierto objeto
TBD	To Be Determinated (Por ser Determinado)