Herramientas de recogida de información sobre proyectos de software libre

Diego Jiménez Jiménez

djimenez@gsyc.escet.urjc.es

Febrero 2005

Visión global

- Disponemos de información pública sobre el desarrollo de proyectos de software libre.
- Por qué no aprovecharla para estudiar y conocer más acerca del proceso de desarrollo del software libre.
- Centrémonos en crear herramientas que, basadas en una interfaz común, extraigan datos de interés de las distintas fuentes de información accesibles . . .
- . . . y almacenen de forma estructurada los resultados para que puedan ser examinados en un futuro.

¿Por qué se publica esta información?

- Proyectos de software libre siguen modelo de desarrollo del bazar:
  - No existe autoridad máxima.
  - Roles de participantes dinámicos.
  - Usuarios se implican de forma activa.
  - Abierto a la incorporación de nuevos integrantes al equipo de desarrollo, adaptados con facilidad gracias a la disponibilidad pública de la información generada por el proyecto.
- La política de disponibilidad permite acceder a (y posteriormente realizar un análisis) ingentes cantidades de información relacionada con el proceso de desarrollo: participantes, ...

Análisis de la información disponible

Objetivos

- Potenciar áreas de la ingeniería del software tradicional.
- Estudio de la evolución del software libre sirve de base de conocimiento para evaluar la salud del proyecto, facilitar la toma de decisiones y pronosticar complicaciones actuales y futuras.
- Analizar modelos que permitan entender el proceso de desarrollo de software libre: interacciones . . .

Fases del análisis

1. Recopilación de cualquier tipo de datos cuantificables desde flujos de información, código fuente, herramientas de desarrollo distribuido, etc.
   Deben ser almacenados en un formato intermedio e independiente de todas las fuentes y herramientas implicadas en ambas fases.
2. Análisis, procesamiento e interpretación de los resultados obtenidos en la fase anterior.

Factor clave: automatización

- Tanto en recogida de datos como en posterior análisis cuantitativo.
- En el pasado: desarrollo de metodologías automatizadas, aunque limitadas a un proyecto concreto.
- En la actualidad: esfuerzos centrados en crear una infraestructura de análisis que integre varias herramientas (que pretenden ser genéricas) capaces de automatizar el proceso al máximo.

  Ventajas :

  - Único esfuerzo en creación de herramientas para analizar un proyecto, adaptarlas a otros supone coste mínimo.
  - Analizar un proyecto con distintas herramientas proporciona una mayor perspectiva sobre el conjunto.

Trabajo realizado

Limitado a la primera de las etapas del análisis.

Parser básico

- Extrae parámetros de interés a partir de un texto, que posteriormente almacena. Potencialmente de cualquier documento, en realidad de ninguno.
- Define interfaz unificada para herramientas futuras.
- Totalmente genérico. NO métodos específicos para obtener o almacenar cada tipo de dato.
- Facilidad de extensión a nuevas fuentes. Necesario adaptar implementación en función del modo de presentación.

Funcionamiento del parser

1. Obtención periódica de documentos.
2. Procesamiento (extracción de datos).
   - Conjunto de patrones predefinidos.
     - Delimitan fragmentos.
     - Reconocen contexto del dato.
   - Selección de fragmentos.
   - Comparación de patrones y obtención de datos.
3. Almacenamiento en formato intermedio (base de datos) de la información no recogida en anteriores pasadas.

Analizadores dedicados

Especializados sobre fuentes de información que representan una interacción entre usuarios y desarrolladores. Todas ellas mantienen un histórico.

- **Foros web en SourceForge**. Recopila datos de un foro completo. Ojo al HTML *variable*.

  Esquema de funcionamiento:

  1. Descarga y parsea página de enlaces a hilos [PE] (en principio la inicial) del foro. Se rescatan enlaces a hilos.
  2. Descarga y examina mensajes de cada hilo de conversación, respetando la jerarquía. Se almacenan los datos obtenidos.
  3. Tras analizar todos los enlaces de PE, recupera link a siguiente PE.
  4. Repetir hasta llegar a última página del foro.

- **Informes de fallos en Debian**. Analiza la secuencia de mensajes (en formato mbox) que llegan al BTS sobre un determinado error.

  Esquema de funcionamiento:

  1. Descarga y parsea mensajes de la página del informe.
  2. Discrimina mensajes según su destino y examina su contenido en busca de pseudocabeceras, etiquetas, etc.
  3. Almacena los resultados obtenidos en sendos análisis.

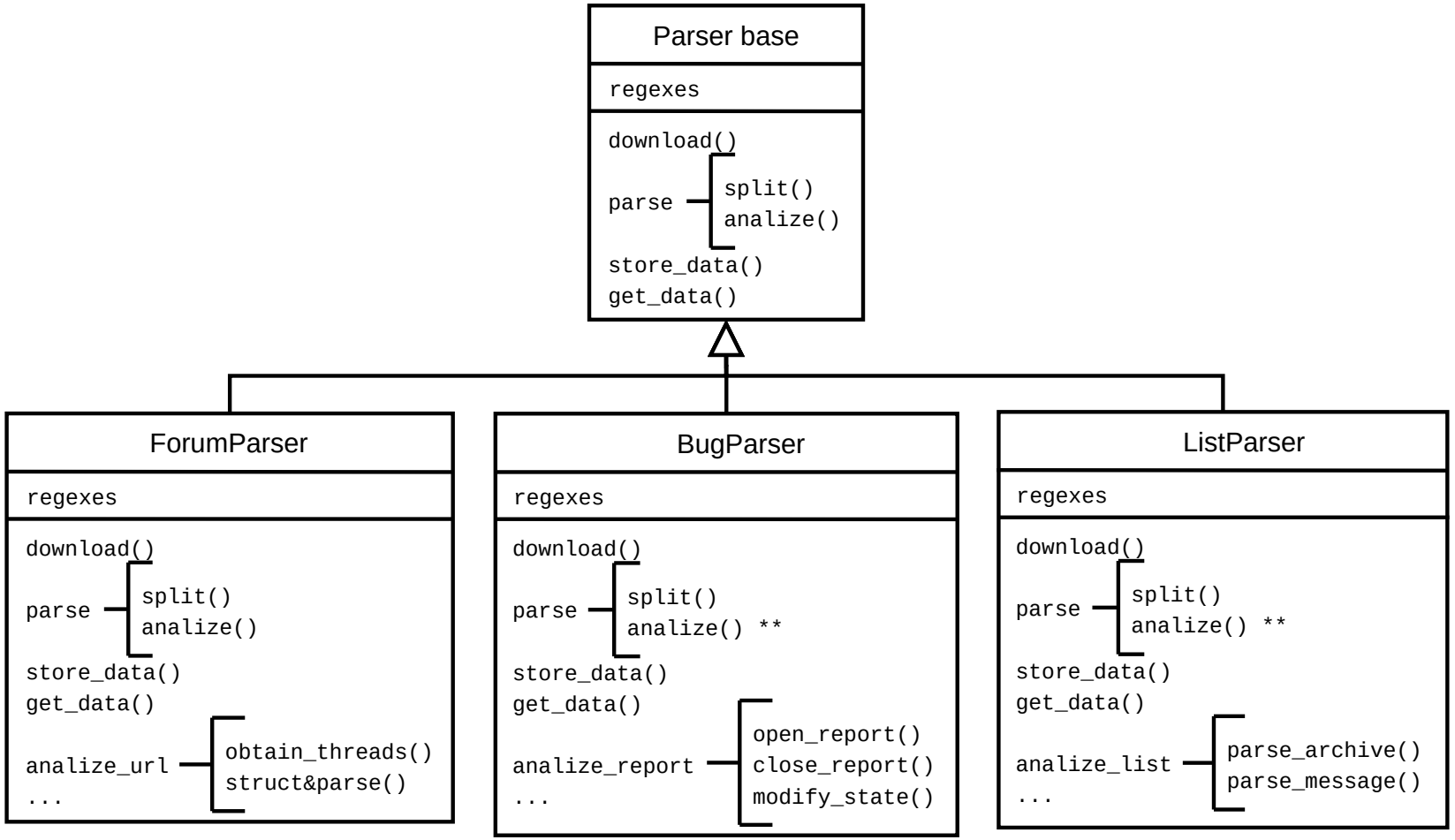
- **Listas de correo de Mailman**. Obtiene datos de los archivos generados desde la creación de una lista de correo.

  Esquema de funcionamiento:

  1. Descarga y parsea página de enlaces a archivos.
  2. Descarga archivo a disco y descomprime.
  3. Parsea los mensajes del archivo, respetando la jerarquía. Se almacenan los datos obtenidos.
  4. Volver a (2) hasta que se agoten enlaces a archivos.

CAPACIDAD DE ACTUALIZACIÓN INCREMENTAL

Estructura general



Parámetros obtenidos

- Foros web
  - Asociados al emisor: nombre, alias.
  - Asociados al mensaje: título, fecha de envío, contenido, identificador del mensaje al que responde, etc.
- Listas de correo
  - Asociados al emisor: nombre, dirección de correo.
  - Asociados al mensaje: identificador, asunto, fecha de envío, etc.
- Informes de fallos
  - Estado del error durante cualquier momento de su existencia: título, versión del paquete, fechas, etiquetas, etc.
  - Asociado al mensaje: emisor, receptor, pseudocabeceras, etc.

Conclusiones

- Prácticamente todo el trabajo puede ser automatizado.
- Obtener conclusiones *definitivas* requiere de estudios que involucren mayor cantidad de datos y de diferentes fuentes. Ahora disponemos de base en que apoyar futuras herramientas que traten la diversidad de fuentes.
- Resultados obtenidos son almacenados en un formato independiente de herramientas de extracción y análisis.

¿ Alguna pregunta ?