

---

*Towards the Integration of CVS  
Repositories, Bug Reporting and Source  
Code Meta-Models*

Alvaro Navarro Clemente  
<[anavarro@gsyc.escet.urjc.es](mailto:anavarro@gsyc.escet.urjc.es)>



# Introducción

## Acoplamiento entre herramientas de análisis de proyectos

- CVS
    - ✓ Cambios
    - ✓ Complejidad
    - ✓ Tamaño
  - Bug Tracking System
    - ✓ Estadísticas sobre fallos en ficheros/módulos
    - ✓ Tasa de solución de fallos
  - Herramienta para análisis del código Fuente
-

# Introducción

- **Problema:** datos de orígenes diferentes
- Representación mediante FAMIX
- Representación básica:
  - Representación de código como entidades (ficheros, funciones..)
  - Cambios y defectos como 'decoraciones'

# Introducción

- Análisis e integración basado en *location by site*:
  - clases, métodos, funciones, cambios y defectos son localizados mediante:
    - Nombre de fichero
    - Número de línea
- Útil para obtener información sobre *MR* y *PR* dentro del código

# Objetivos

- Extender el modelo FAMIX con nuevos meta-modelos.
  - Diseñar una arquitectura que integre la obtención de datos de orígenes diferentes.
  - Análisis dinámico a nivel de código y metadatos
  - Aplicar el modelo a Mozilla
-

# Arquitectura

- Punto de encuentro común para todos los orígenes de datos.
  - CVS y Bugzilla
    - *Release History database*
    - Almacena cambios y defectos relevantes
  - Código Fuente
    - Uso de RSF para representar modelos como clases, métodos, tipos y funciones
    - Parseo del código mediante 2 fases:
      - Primera fase de configuración y modelado en AOL
      - Representación final en RSF
-

# Arquitectura

- Integración final aún en fase de desarrollo
- Idea de representar todo en FAMIX:
  - Consultas a la base de datos y posterior conversión en RSF
  - Integrar resultados de código y la consulta anterior en RSF

# Herramientas

- Compiler Wrappers
- Extractores información de C++
- Conversores a formato RSF atendiendo al estándar FAMIX

# Caso de estudio: Mozilla

Desarrollado en C/C++, XML y HTML

- *10,000 ficheros con un total de 3,7 millones de LOCs*
  - *90 módulos mantenido por 50 desarrolladores*
  - *180,000 PRs en Bugzilla*
  - *430,000 MRs en el repositorio CVS*
  - *“Si un lenguaje contiene una característica peculiar seguramente será usado a pesar de su impacto a la hora de entender, mantener y portar el código”*
  - *Conclusiones sobre el código (estructura de datos, funciones..)*
-

# Conclusiones

- La integración de análisis heterogéneos permiten obtener conclusiones sobre el proyecto a distintos niveles de abstracción