

Hackfest de ATK/AT-SPI2

Motivos, organización y conclusiones

```
static void
properties(GObjectClass
*gobject_class)
{
    mSpec *pspec;
```

```
attribute */
uint64
CODE,
ode.",
ode",
0,
64,
/*
/,*
E
```





Introducción

Que necesita un AT?

- Imaginar que quereis hacer una nueva AT
- ¿Que cosas necesitais?
- ...

Un AT necesita cosas como ...

- Saber que aplicaciones están disponibles ...
- Saber cuál está activa ...
- Poder conectarse a cambios en estas aplicaciones ...
- Poder interactuar con ellas ...

En general ...

- Todo eso se puede resumir como que necesita un servicio proporcionando esa información
- No eres el único AT por aquí ...
- De manera homogénea

AT-SPI

- AT-SPI es este servicio, un proceso aparte
- Para acceder a este servicio necesitas un IPC
 - at-spi usaba CORBA, at-spi2 usa DBUS
- La manera más eficiente es usar algun API (librería binding)

¿Y qué necesita AT-SPI2?

- AT-SPI2 tiene que exponer esa información a las aplicaciones
- Pero como la obtiene?
- Esto parece fácil: pues usar DBUS!

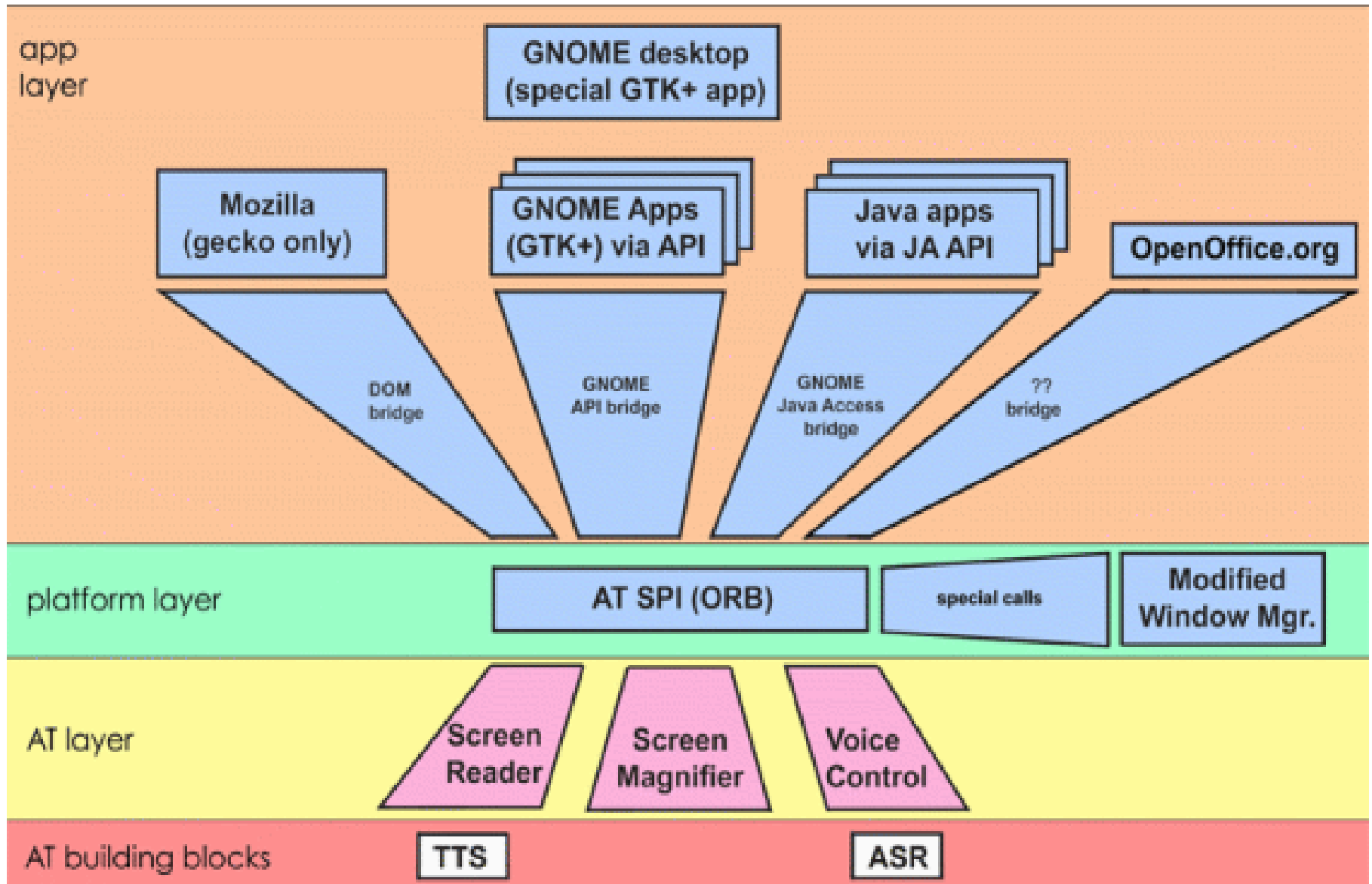
Pues empezemos mandar mensajes por DBUS

- A nivel de aplicación, no parece tener mucho sentido
- Pues en cada toolkit: Gtk+, Clutter, Gecko, Webkit, ...
- Parecen muchos, no parece que tenga sentido implementar la comunicación por DBUS en cada uno de ellos

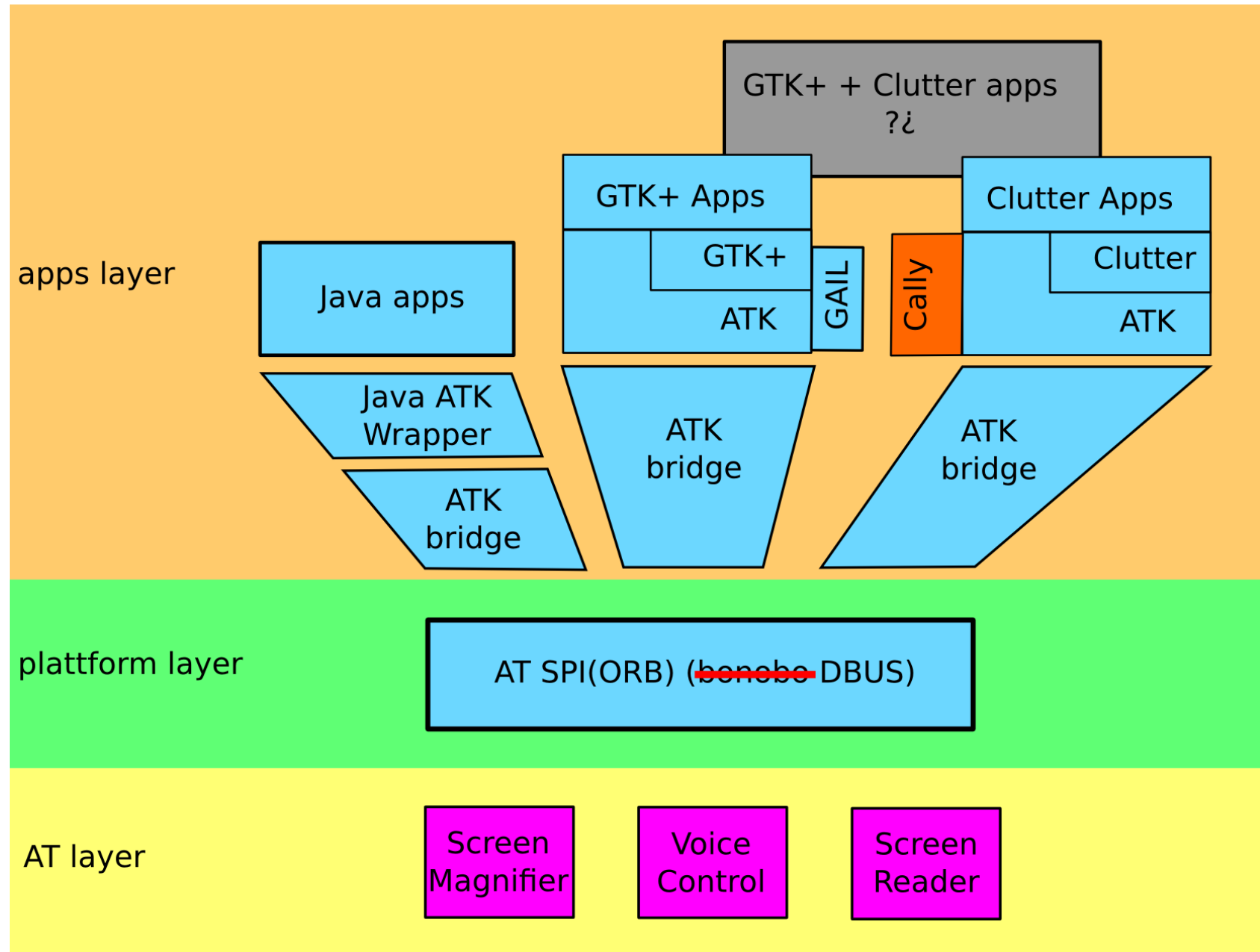
Mejor abstraerse un poco

- ATK es una abstracción de un UI toolkit
- Incluye un objeto accesible y una serie de interfaces.
- En cada toolkit los distintos widgets proporcionarían un objeto ATK
- AT-SPI2 proporciona un bridge ATK, utiliza este objeto ATK para interactuar con el widget final

Arquitectura a11y en GNOME 2.0



Architecture en GNOME 3.0



Interfaces en ATK

- Las aplicaciones pueden comprobar que interfaces ATK implementa un objeto
- Las interfaces permiten a las ATs hacer heurísticas funcionales
- La implementación de a11y para un widget debe decidir que interfaces expresa sus capacidades
- Ejemplos:
 - AtkComponent
 - AtkAction
 - AtkText

Quien implementa ATK? Todos?

- Hay implementaciones de ATK para Gtk+, Clutter, Gecko, Webkit, Unity (empezando) y otros
- Aplicaciones JAVA no tienen una implementación de ATK, utilizan un puente propio con at-spi y un wrapper con at-spi2
- Qt ha preferido crear un puente específico (qt-bridge)

Freedesktop

- El último punto es importante
- AT-SPI2 es el estándar de accesibilidad para Linux
- KDE está intentando empezar a usar esta tecnología también

¿Para qué un
hackfest?

Ya van 10 años ...

- ATK/AT-SPI empezó su existencia hace unos 10 años
- Aunque se ha ido actualizando su API ha sido más o menos la misma
- La migración a DBUS y GNOME 3.0 absorbió gran parte del esfuerzo

Hay cosas por mejorar ...

- En el 2010 se organizaron dos hackfests de accesibilidad
- La tarea principal era preparar la llegada de GNOME 3.0
- Pero se comentó que los dos necesitarían ser actualizados, planeando una rotura de API

El hackfest

Coordinación previa

- Se usó el bugzilla y la gente incluía las propuestas allí:
 - https://bugzilla.gnome.org/show_bug.cgi?id=638537
- Se reservaba un tiempo en las reuniones semanales de accesibilidad
- Listas de correo

Día 1

- Creación de una agenda para la semana basados en los bugs
 - Aprox 40 bugs
- Clasificación por temas
 - <https://live.gnome.org/Hackfests/ATK2011/Agenda>
- Discusiones iniciales

Día 2

- IA2:
 - Discutir que cosas podrían ser útiles en ATK
 - Mirar incluso de las propuestas de Mozilla para IA2
- Gestión del foco
 - Proporcionar en AT-SPI una forma de preguntar por el objeto con el foco
 - Simplificar los eventos relacionados y el tracking

Día 2 (cont)

- Eventos
 - Notify vs Property-change?
 - No estaba claro la necesidad de property-change
 - Cual era su objetivo?
 - Realmente se puede usar?
 - Incluir no solo el nuevo valor en el evento
 - Más señales de selección de texto?

Día 3

- Eventos
 - Discutir nuevas adiciones de eventos
 - Filtrado de eventos?
 - Ser más restrictivo con el registro global de eventos
- Jerarquía de objetos en elementos comunes como menus, combo boxes, árboles, ...
- Definición de roles

Días 4 y 5 (por venir)

- Eventos de teclado
- Revisión de las cosas que están rotas
- Documentación
- RelationSets?

Días 4 y 5 (cont)

- Buscar una manera agnóstica de determinar que Ats se estan
- Revisión de AtkUtil
- Rendimiento
- Releasing

Conclusiones

Conclusiones

- El hackfest aún no ha terminado! ;)
- El hackfest es la oportunidad para reunir a gente con experiencia en accesibilidad, tanto usuarios de ATK como implementadores de ATK
- Durante los primeros días ya se han llevado a cabo una gran cantidad de discusiones y trabajo

Referencias

- <http://live.gnome.org/Accessibility/>
- <http://live.gnome.org/Accessibility/GNOME3>
- <http://projects.gnome.org/accessibility/>



PREGUNTAS?