



Nuevas funcionalidades de localización para gvSIG Mobile 1.0 basadas en libLocation

Juan G. Jordán, Institut de Robòtica, UV
Manuel Planells, Institut de Robòtica, UV

Institut de Robòtica UNIVERSITAT VALÈNCIA (📱📶)



Objetivos

Soporte a las plataformas J2SE y J2ME CDC

Proporcionar funcionalidad tanto en el ámbito de la navegación como de la comunidad GIS

Soporte a diferentes protocolos y sistemas de localización

Soporte de almacenamiento de datos para waypoints, tracks y rutas

Interfaz de programación sencilla y de alto nivel

Extensibilidad



Metodología

Uso de JSRs

Uso de Java ante soluciones JNI

Reutilización de código

Optimización de recursos

Uso extensivo de tests unitarios

/**

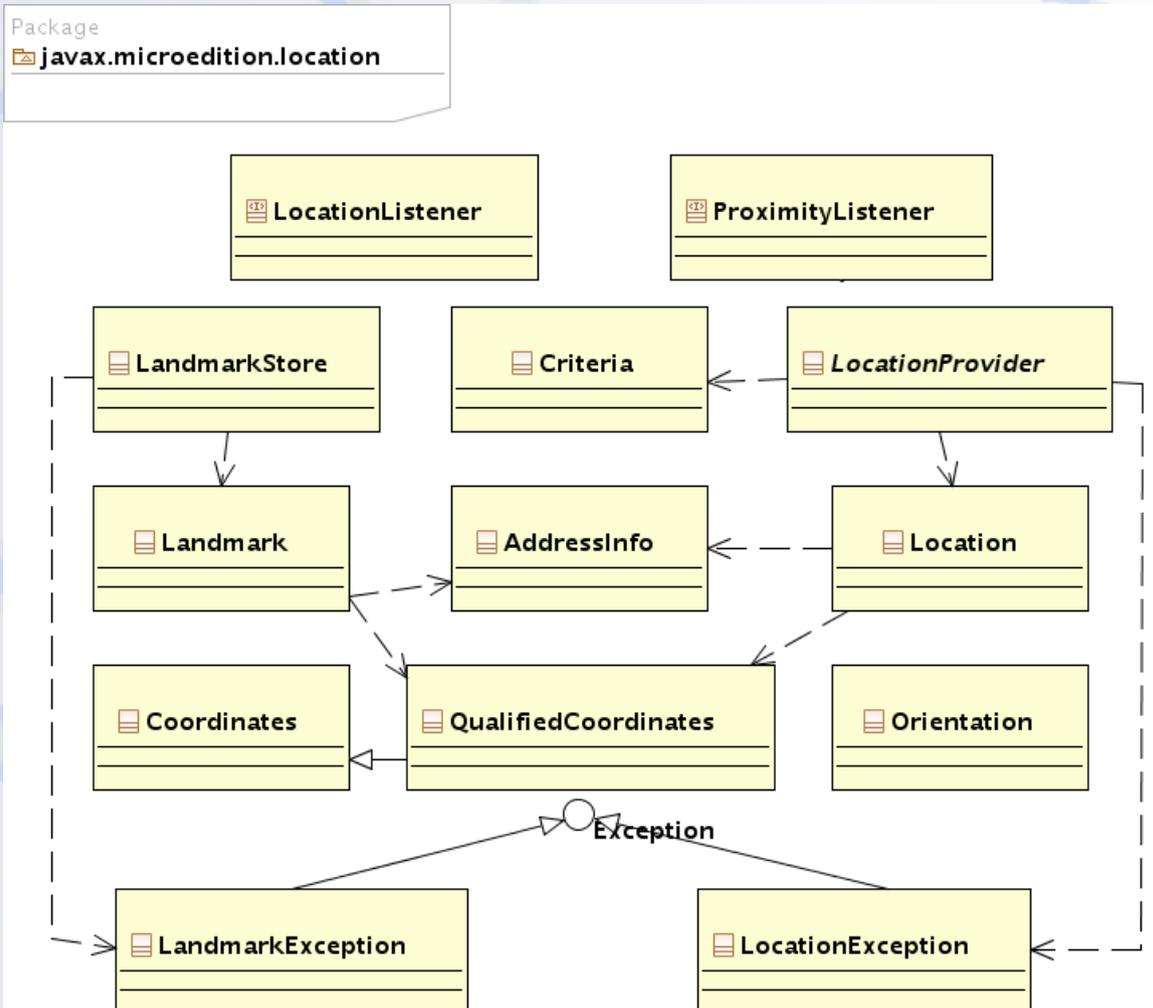
* Comment As You Code

*/

Metodología de desarrollo recomendada para gvSIG 2.0



La librería



“Partiendo”
de la API de la
JSR 179 de
Localización...

LocationProvider

API JSR 179

=====

```
static addProximityListener()

static getInstance()
static getLastKnownLocation()

getLocation()

getState()

static removeProximityListener()
reset()
setLocationListener()
```

API libLocation

=====

```
addAreaListener()
addLocationListener()
addMovementListener()
addPropertyChangeListener()
addProximityListener()
getCurrentTimestamp()
---
---
getLastLocation()
getLocation()
getLocationProperty()
getState()
removeAreaListener()
removeMovementListener()
removeLocationListener()
removePropertyChangeListener()
removeProximityListener()
reset()
---
setMovementListener()
removeMovementListener()
startAveraging()
stopAveraging()
cancelAveraging()
```



Cambios en la interfaz



Se eliminan los métodos estáticos o sustituyen por métodos de instancia

Desaparece el método de factoría getInstance()

Se ha creado un gestor de proveedores de localización (LocationProviderManager)

LocationListener deja de ser único

setLocationListener() → addLocationListener()

Se han añadido nuevos listeners



Comunicaciones

Los providers reciben flujos de texto plano o binarios de una clase que procesa indistintamente cualquier tipo de Input Stream

libLocation					
Provider NMEA	Provider NMEA	Provider NMEA	Provider SiRF	Provider TSIP	Provider GPSd
RXTX GCF	JSR182 <small>javax.microedition.bluetooth</small>	File <small>java.io</small>	File <small>java.io</small>	RXTX GCF	HTTP
GPS NMEA interno/serie	GPS NMEA bluetooth	GPS NMEA simulado	GPS SiRF simulado	GPS TSIP interno/serie	GPS demonio GPSd



Flexibilidad y extensibilidad

Cada provider define un decodificador general

Debe ser extendido para decodificar cada tipo de mensaje

En tiempo de ejecución se registran los decodificadores que se desea utilizar

Los mensajes que no tienen un decodificador asociado se ignoran

Registro de proveedores



LocationProviderManager

```
=====
setConfigurationParameters()
discoverPlugins()
getDescription()
getLastKnownLocation()
getLocationMethod()
getLocationProvider()
getLocationProviderNames()
isLocationMethodSupported()
registerLocationProvider()
unregisterLocationProvider()
unregisterAll()
```



Registro de proveedores

En tiempo de ejecución se registran los proveedores que estén disponibles

Pueden registrarse proveedores de otros ficheros JAR (plugins)

Se dispone de métodos para interrogar sobre los proveedores registrados, sus propiedades y para la instanciación de éstos

Se dispone de métodos de configuración de parámetros de los proveedores



Listeners

JSR 179 Location API

LocationListener

ProximityListener

JSR 293 Location API 2.0

AreaListener

Propios de libLocation

AveragingListener

MovementListener

PropertyChangeListener



LocationListener

Eventos de localización y estado

Intervalo de actualizaciones y edad máxima de las muestras

Cambios de estado del proveedor

AVAILABLE / TEMPORARY_UNAVAILABLE / OUT_OF_SERVICE

LocationProvider

=====

```
void addLocationListener(LocationListener listener, int interval,  
    int timeout, int maxAge)  
void removeLocationListener(LocationListener listener)
```

LocationListener

=====

```
void locationUpdated(LocationProvider provider, Location location)  
void providerStateChanged(LocationProvider provider, int newState)
```



ProximityListener

Eventos de proximidad a coordenadas

Típica aplicación: paso por waypoint

LocationProvider

=====

```
void addProximityListener(ProximityListener listener, Coordinates  
    coordinates, float proximityRadius)  
void removeProximityListener(ProximityListener listener)
```

ProximityListener

=====

```
void proximityEvent(Coordinates coordinates, Location location)  
void monitoringStateChanged(boolean isMonitoringActive)
```



AreaListener

Entrada o salida en un perímetro geométrico

El listener previsto por la JSR293 no informa si se está entrando o saliendo, sólo que se cruza el perímetro :(

LocationProvider

=====

```
void addAreaListener(AreaListener listener, GeographicArea area,  
    Criteria criteria, int interval)  
void removeAreaListener(AreaListener listener)
```

AreaListener

=====

```
void areaEvent(GeographicArea area, Location location)  
void monitoringInfo(int locationMethod, int interval)  
void monitoringStateChanged(boolean isMonitoringActive)
```

AveragingListener

Promediado de muestras

Posibilidad de filtrar muestras: por precisión horizontal, vertical y por desviación de la media

El proceso se puede detener y/o cancelar

LocationProvider

=====

```
void startAveraging(AveragingListener listener, int maxSamples,  
    int filterOpts)  
void stopAveraging()  
Void cancelAveraging()
```

AveragingListener

=====

```
void averagingFinished(Location average, int samples, int  
    filteredHAcc, int filteredVAcc, int filteredDev)  
void averagingProcess(Location partialAverage, int percent)  
void averagingCancelled()
```



MovementListener

Filtrado de coordenadas que aportan poca información del movimiento del dispositivo

Basado en cambios en la distancia, velocidad y orientación

Un evento para coordenadas seleccionadas y otro para coordenadas eliminadas, y el motivo de su filtrado

LocationProvider

=====

```
void setMovementListener(MovementListener listener, double minDist,  
    double minSpeed, double minCourse)  
void removeMovementListener()
```

MovementListener

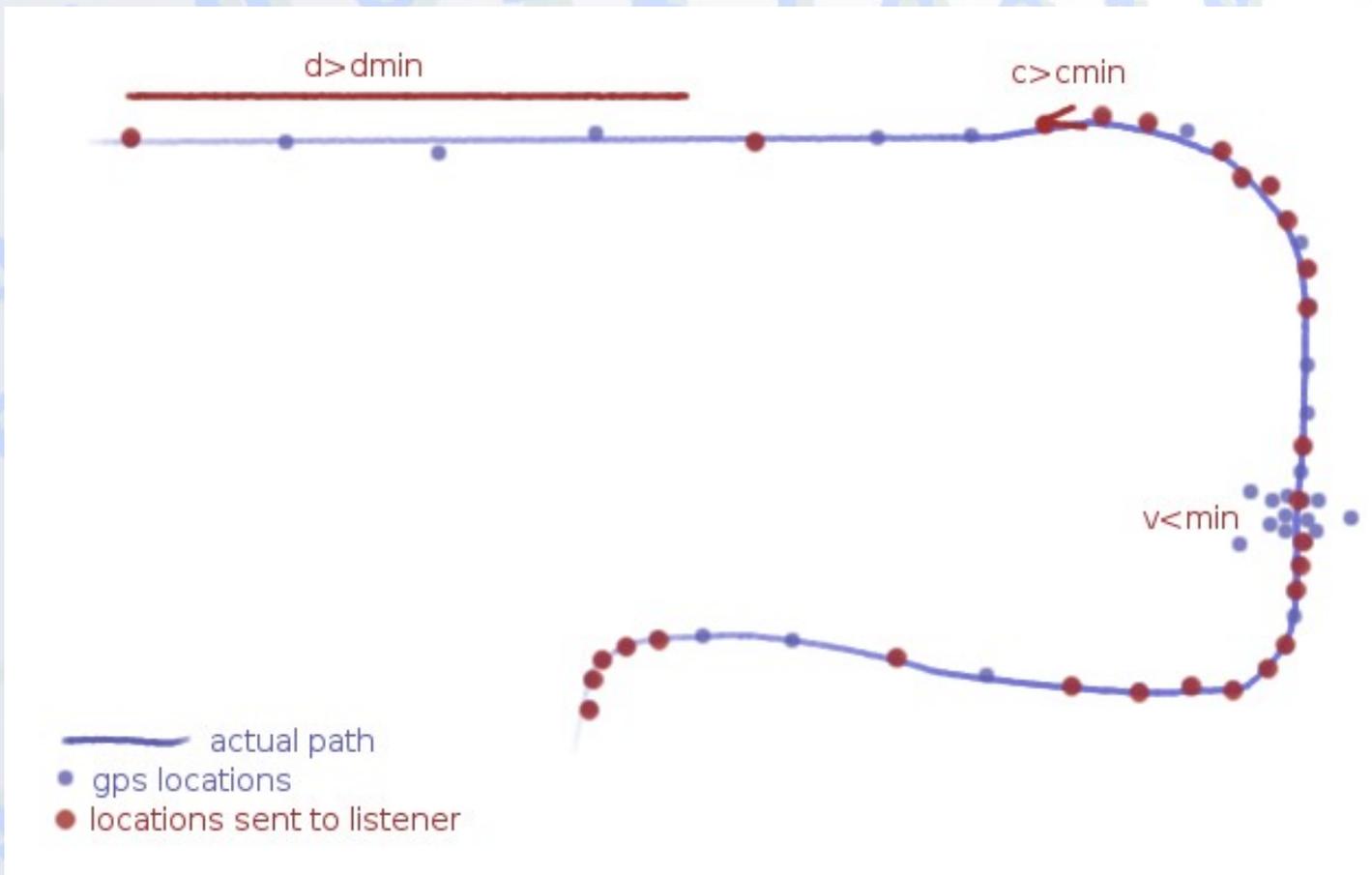
=====

```
void movingLocation(Location location)  
void discardedLocation(Location location, int filter)
```

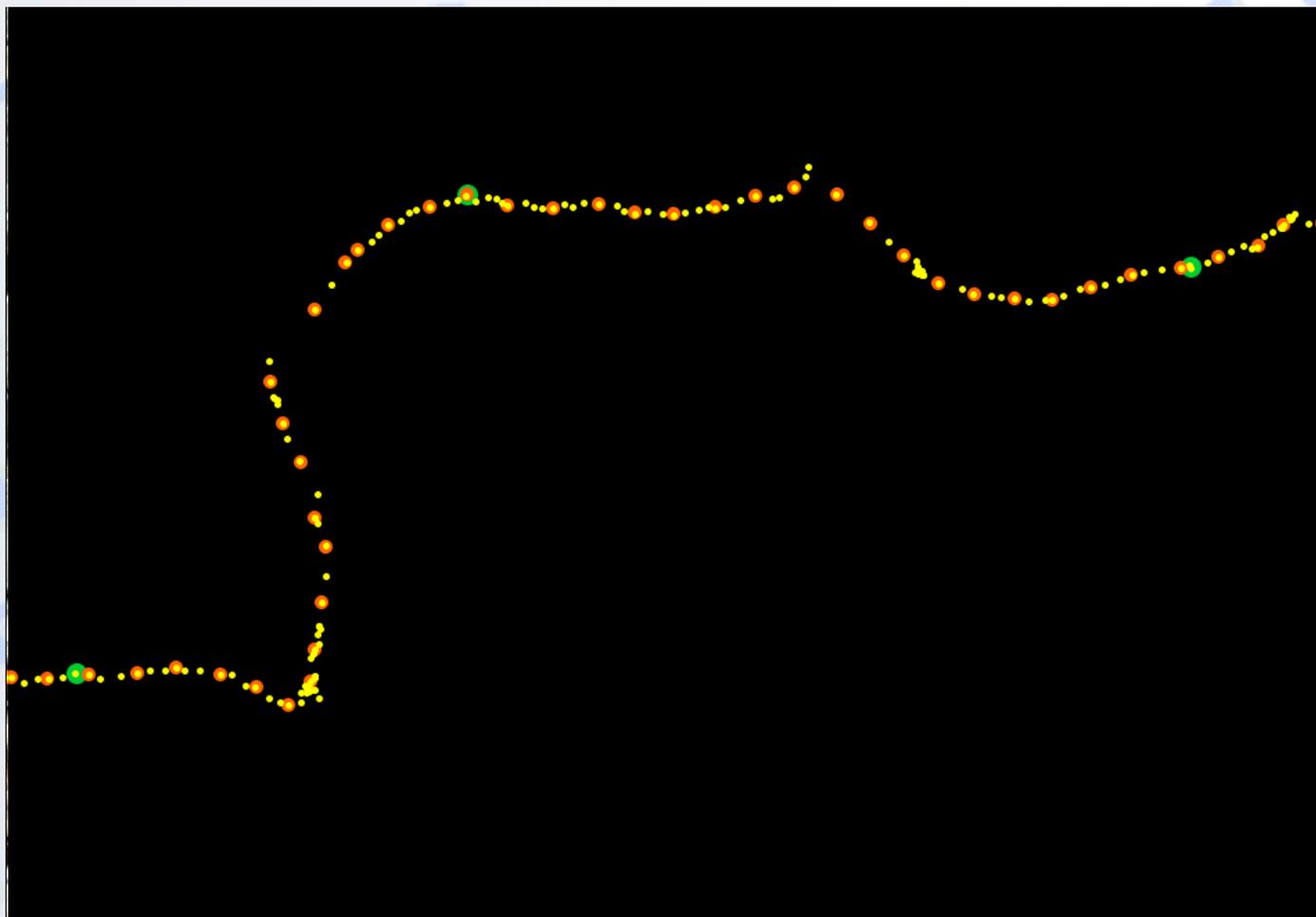


MovementListener

Ideado para el suavizado de trazas



Ejemplos de suavizado



PropertyChangeListener

Suscribirse a cualquier propiedad enviada por el proveedor de localización

LocationProvider

=====

```
void addPropertyChangeListener(String key, PropertyChangeListener listener)
void addPropertyChangeListener(PropertyChangeListener listener)
void removePropertyChangeListener(String key, PropertyChangeListener listener)
void removePropertyChangeListener(PropertyChangeListener listener)
```

PropertyChangeListener

=====

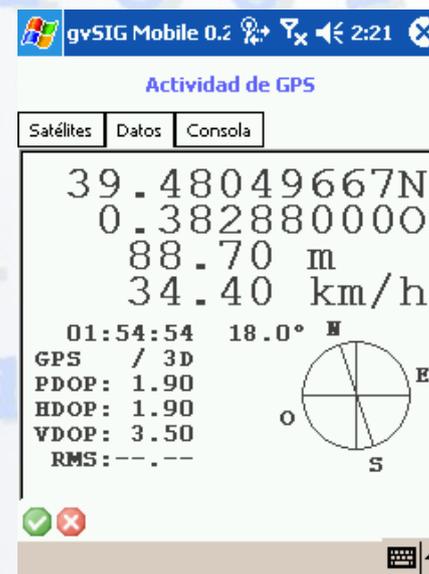
```
void propertyChange(PropertyChangeEvent evt)
```

PropertyChangeListener

Actualización de propiedades no soportadas directamente por la API JSR 179

constelación de satélites, PDOP, HDOP, HPE, VPE, EPE...

propiedades que envíen nuevos proveedores



LandmarkStore, TrackStore

Gestión de waypoints, rutas y tracks

Funciones de búsqueda y filtrado de landmarks
(waypoints) y tracks

Importación y exportación a GPX y CSV

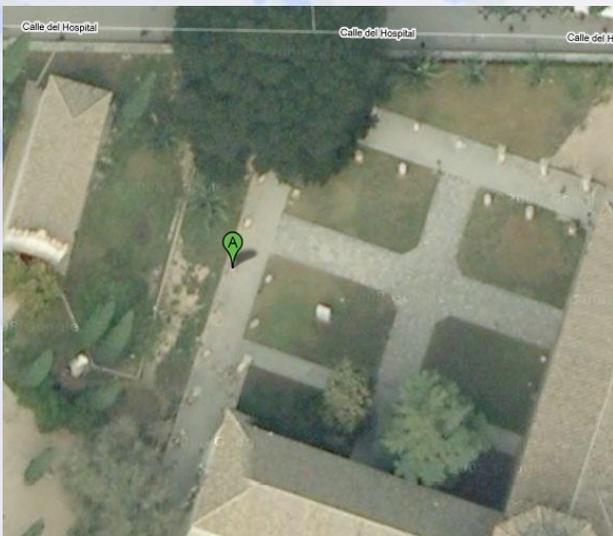
Implementación actual basada en HSQLDB



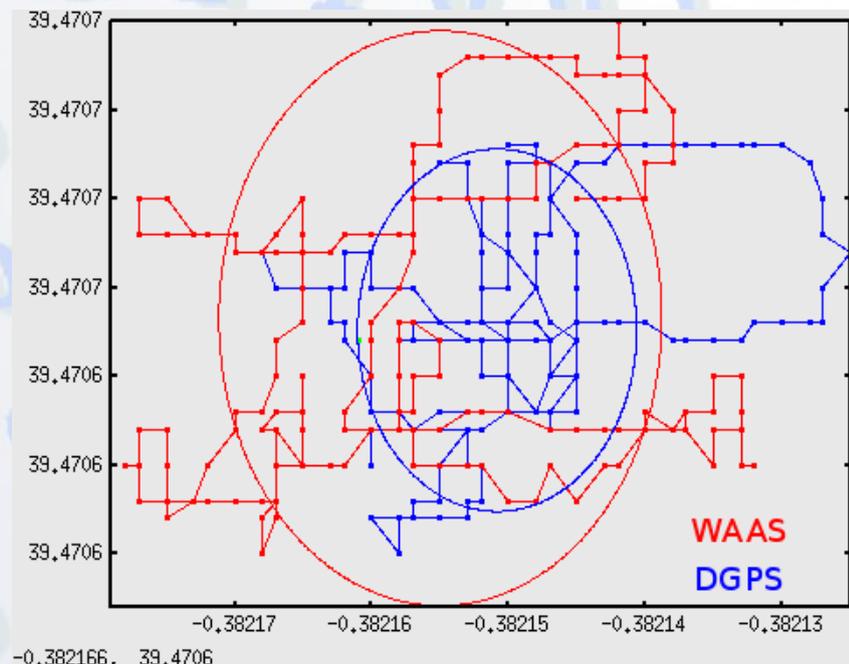
Otras funcionalidades

Conexión a servidor NTRIP y reenvío de correcciones RTCM a dispositivo GPS (DGPS)

Pruebas NTRIP realizadas



Ubicación aproximada
Promediado durante 5 min



GPS autónomo

Error de varios m

WAAS (EGNOS)

RMS 1.7m

DGPS (RTCM)

RMS 1.2m

Otras funcionalidades

Registro de observables en ficheros RINEX para la aplicación de correcciones en postproceso



2.11	OBSERVATION DATA								M (MIXED)	RINEX VERSION / TYPE
GPSTime 2.70 3641	Rinex Merge								29-Sep-09 02:27:42	PGM / RUN BY / DATE
BURG										MARKER NAME
19314M001										MARKER NUMBER
ITACyL	ITACyL									OBSERVER / AGENCY
4624K01644	TRIMBLE NETR5								NP 70.55	REC # / TYPE / VERS
0										RCV CLOCK OFFS APPL
30278353	TRM55971.00									ANT # / TYPE
4712044.5241	-303624.8272	4274801.4150							APPROX POSITION XYZ	
0.0000	0.0000	0.0000							ANTENNA: DELTA H/E/N	
1 1 0									WAVELENGTH FACT L1/2	
8 C1 P1 P2 C2 L1 L2 S1 S2									# / TYPES OF OBSERV	
15.000									INTERVAL	
2009 9 26 0 0 0.0000000 0	GPS								TIME OF FIRST OBS	
END OF HEADER										
09 9 26 0 0 0.0000000 0	17R07G25G17G20G23G13R08R23R09R24G31G04									
G11R10G02G07G32										
20577264.688	20577263.305	20577266.586						2554676.30906		
1986459.19707	47.000	42.000								
22884801.766									-10523274.16406	
-8170982.92007	45.000	37.000								
22261474.445									22261469.500	-15495704.19506
-12044766.52707	47.000	42.000								
21638381.398									-19268635.20206	



Estado de desarrollo

A nivel de librería el desarrollo ha cubierto los objetivos. A completar:

Corrección de bugs

Terminar algunas funcionalidades extra

Desarrollados 4 proveedores de localización: NMEA 0183, SiRF, TSIP y GPSd

A nivel de integración con gvSIG Mobile —a través de la extensión GPS— el estado de desarrollo es del 60%. A completar:

Diálogo de actividad GPS

Diálogo de propiedades GPS

Gestión de waypoints y tracks

Navegación a través de waypoints y gestión de alertas



Proveedor NMEA

Protocolo soportado por casi
el 100% de GPS

GPS de consumo

Velocidad binaria lenta

Usuarios no profesionales

PVT

Altitud, orientación

Satélites

HDOP, VDOP, HPE...

Mensajes
implementados

GPDBT

GPGGA

GPGGL

GPGSA

GPGST

GPGSV

HCHDG

GPRMC

PGRME

GPVTG



Proveedor SiRF

Chipsets SiRF Star II y III

GPS de consumo

Velocidad binaria > 19200

Usuarios medios y avanzados

Lo mismo que NMEA y además...

Estado avanzado del receptor

Observables: pseudorrango, fase de portadora, etc.

Posibilita el almacenamiento en formato RINEX para el postproceso

Mensajes

implementados

2. Measure Navigation Data Out

4. Measure Tracker Data Out

27. DGPS Status Format

28. Navigation Library
Measurement Data

41. Geodetic Navigation Data

50. SBAS Parameters



Proveedor TSIP

Chipsets Trimble

GPS de alta gama

Velocidad binaria elevada

Usuarios profesionales

Lo mismo que NMEA y además...

Estado avanzado del receptor

Observables: pseudorrango, fase de portadora, dopler, etc.

Posibilita el almacenamiento en formato RINEX para el postproceso

Implementación básica

Mensajes implementados

0x84. Double-Precision LLA
Position Fix & Clock Bias



Proveedor GPSd

Todo tipo de GPS usando
diversos protocolos:

NMEA, SiRF, TSIP, Garmin,
EverMore, Navcom,
Rockwell/Zodiac y uBlox

GPS de todas las gamas

Es un demonio de Linux que
permite compartir GPS por
varias aplicaciones

Una opción para conectar con
dispositivos no soportados
directamente por libLocation

Basado en el
“viejo” protocolo
GPSd

Mensajes implementados

- O. Informe de tiempo + posición +
velocidad
- Y. Informe de la constelación de
satélites
- X. Informe de estado del
dispositivo



Trabajo futuro

Completar integración con gvSIG Mobile

Mejoras sobre la librería

Autoconfiguración de proveedores de localización

Desarrollo de un gestor de las comunicaciones

Aplicación de correcciones diferenciales por software en tiempo real (DGPS y/o RTK)

Almacenamiento de datos basado en DAL (librería de acceso a datos para gvSIG 2.0)



Gracias por su atención ;)



Descarga del repositorio de gvSIG Mobile

<https://gvsig.org/svn/gvSIG-mobile/libLocation/trunk/libLocation/>

Contacto

jjordan@robotica.uv.es

