AGL Navi-EG
All member meeting @ Dresden

Yoshito Momiyama
2017.10.19
About Myself

- I’m Yoshito Momiyama, working at AISIN AW CO., LTD.
- I work in the Vehicle Information Technologies department, specifically in the Software Fundamental Technology Group.
- I have developed Navigation applications for 11 years.

Hello, everybody. Sorry, I am traveling with my family. I would like Olivier-san and Yamaguchi-san, to present to everybody.
About Myself

- I’m Olivier Pirlet, working at AISIN AW for 17 years.
- I work in the Vehicle Information Technologies department.
Contents

- About Navi-EG
- AGL Navigation Architecture Update
  - Map Drawing Architecture
  - AGL Application Framework Binding
- AGL Navigation API
  - API Strategy
  - Explanation of GENIVI API
  - AGL Extended
- Future tasks
Contents

- About Navi-EG
  - AGL Navigation Architecture Update
    - Map Drawing Architecture
    - AGL Application Framework Binding
  - AGL Navigation API
    - API Strategy
    - Explanation of GENIVI API
    - AGL Extended
- Future tasks
Navi-EG Members

Expert Group Leader:
・Yoshito Momiyama

Expert Group Members:
・Takuo Koguchi
・Seiji Goto
・Kentaro Yamauchi
・Masaya Hashida
・Takeshi Hoshina
Navi-EG Activities

- **Members Grows**
  - TELNAV have joined Navi-EG

- **Discussions**
  - We held a telephone conference once every two weeks
  - We are discussing the following contents
    - AGL Navigation architecture
    - AGL Navigation API
    - Open source license of Navigation API
    - Development items

New Member
Navi-EG Goal

AGL Apps.

- Navi Application
- POI Application

AGL Defined API

- GENIVI-API
  - Extended by Navi-EG

AGL Compliant Navi engine

- OSS Navi (GPS-navi)
  - Selectable
- Product Navi

AGL Device API

Devices/Middleware

(1) AGL provides common Navi API.

(2) AGL scalability should be allowed as long as compatibility allows.

(3) AGL provides P/F which allows to select OSS or Product Navi.

(4) AGL provides P/F which allows to select OSS or Product Navi.
Activity Results

Members demos

AGL official demos
- OSS navigation
- POI setting application
- Cluster Map view Application

Members demos
- Product US navigation on AGL (B.B.)

Members demos
- Product JP navigation (in Denso demo space)

Members demos
- Product JP navigation on AGL (C.C.)
  - It can voice guide using GPS.
Contents

- About Navi-EG

- AGL Navigation Architecture Update
  - Map Drawing Architecture
  - AGL Application Framework Binding

- AGL Navigation API
  - API Strategy
  - Explanation of GENIVI API
  - AGL Extended

- Future tasks
In foreground
- Route Guidance (with voice sound).
- Map drawing including many information (traffic information, POI, etc...).
- Updating of the current position.
- Re-routing by incident.

Navigation needs to do a lot of draw processing.

In background
- Route Guidance (by voice sound).
- Updating of the current position.
- Re-routing by incident.

Navigation need to keep running always in the background. And it need to be able to output voice sound.
Architecture Update

Draw processing models

- NaviCore drawing model

Navi Application

Application

Framework

Control

Process boundary

Navicore

Drawing Map

Composition by window manager

SANDS AVE

Composition by window manager

©AISIN AW CO., LTD. All Rights Reserved.
**Architecture Update**

**Draw processing models**

- Framework drawing model

---

- Navi Application
  - Application
  - Framework

- Process boundary

- Control
  - Map Data

- Drawing
  - HMI and Map

- Navicore

---

**Navicore**

- SANDS AVE
Architecture Update

Draw processing models

- Image download model

Navi Application

Application Framework

Process boundary

Control

Map Image

Navicore

Drawing HMI on Map Image

SANDS AVE
## Architecture Update

### Draw processing models

#### Comparison of models

<table>
<thead>
<tr>
<th></th>
<th>Navicore drawing</th>
<th>Framework drawing</th>
<th>Image download</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Response</td>
<td>Good</td>
<td>Bad (A large amount of data needs to be sent)</td>
<td>Bad (Image data needs to be sent)</td>
</tr>
<tr>
<td>Scalability</td>
<td>Bad (Unable to process on external server)</td>
<td>Bad (Difficult to process on external server)</td>
<td>Good</td>
</tr>
<tr>
<td>Customizability</td>
<td>Normal (Can customize by changing Navicore)</td>
<td>Good (Can customize by application)</td>
<td>Normal (Can customize by changing Navicore)</td>
</tr>
<tr>
<td>Usage</td>
<td>GENIVI</td>
<td>OSM apps</td>
<td>Google Map OSM apps</td>
</tr>
</tbody>
</table>

We have chosen this considering a slow response AGL security framework.
Our requirement

- The window manager can bind the surface of the service to the surface of the application.
- When making the application display state, the window manager also makes the bound surface transition to the display state at the same time.
- The application manager should not stop the navi core even when the application goes off screen.
Contents

- About Navi-EG
- **AGL Navigation Architecture Update**
  - Map Drawing Architecture
  - **AGL Application Framework Binding**
- AGL Navigation API
  - API Strategy
  - Explanation of GENIVI API
  - AGL Extended
- Future tasks
AGL API implementation have to use afm-binder

Communication between applications and binders uses http or web-socket. An API is implemented using either REST over HTTP or a kind of JSON RPC.

afb-daemon becomes the proxy of the API. The communication protocol between Binder and service is not restricted. (Such as d-bus, sunrpc, etc.)
AGL Navigation architecture concept.

- AGL Apps.
  - Navi Application
  - POI Application

- AGL Defined API
  - AGL Navigation API

- AGL Compliant Navi engine
  - OSS Navi (GPS-navi)
  - Product Navi

Selectable
AGL Navigation architecture of API binding.

- Navi Application
- POI Application

AGL Navigation API
All bindings implement the same AGL Navigation API protocol.

Binding for OSS Navi
Binding for Product Navi

web-socket with Json. (Candidate)

OSS Navi (GPS-navi)
Product Navi

API of each NaviCore
There is no constraint on NaviCore native API.
AGL Navigation architecture of API binding.

- Navi Application
- POI Application

web-socket with json. (Candidate)

Binding for OSS Navi
OSS Navi adaptation

Binding for Product Navi
Product Navi adaptation

afb-daemon

D-bus

OSS Navi (GPS-navi)

Any

Product Navi

Navi EG performs a common implementation of the application side protocol. Using this implementation is a condition to certify the AGL Navi API. This ensures API compatibility.

Navi vendors develop and provide Navicore and Binding as a set.
Contents

- About Navi-EG
- AGL Navigation Architecture Update
  - Map Drawing Architecture
  - AGL Application Framework Binding
- AGL Navigation API
  - API Strategy
  - Explanation of GENIVI API
  - AGL Extended
- Future tasks
A lot of navigation APIs are defined

- GENIVI Navigation API
  - https://at.projects.genivi.org/wiki/display/NAV/IVI+Navigation+Home
- Google Map API
  - https://developers.google.com/maps/documentation/directions/?hl=en
- Qt Location API
- Each vendor specific API

We do not want any more API fragmentation. In principle, the AGL API conforms to GENIVI Navigation API.
The GENIVI Navigation API is described in dbus XML.

AGL API needs to be implemented using afm-binder.

We correspond to afm-binder by converting dbus XML to json.

The AGL Navigation API targets native applications at first. HTML 5 will not be covered for the time being. Because there is a possibility that the W3C API is better in HTML5.

A afm-binder can use only http or websocket, it has performance problems. If performance is bad, binary data have to flow through afm-binder.
We will implement
- AGL Navigation API binding source code
  - It’s common source of all navigation core.
- AGL Navigation API adaptation for OSS Navi
  - It’s reference implementation of the navigation API binding.
- Navigation API stub for API testing
  - It’s reference implementation of the navigation API response.
  - In our plan, OSS Navi are using evaluation to the AGL Navigation API and afm-binder.

Navi application and OSS NaviCore are using GPS navi.
Contents

- About Navi-EG
- AGL Navigation Architecture Update
  - Map Drawing Architecture
  - AGL Application Framework Binding
- AGL Navigation API
  - API Strategy
  - Explanation of GENIVI API
  - AGL Extended
- Future tasks
There is the following for a main function of GENIVI NaviAPI and is included in AGL NaviAPI.

- **Mapview**
  - 2D/3D map draw, change scale, change direction
  - route/mark draw on map
  - follow current position

- **Route**
  - set destination, waypoints
  - set route preferences
  - calculate route, alternative route

- **Guidance**
  - start/stop guidance
  - re-voice
  - signal of on/off road, on/off route, arrival destination
AGL NaviAPI spec1.0
included GENIVI NaviAPI(2)

- Location input

- mapmatched position
  - start/stop simulation
  - set/get current position

- configuration
  - set measurement, language
  - set time format, coordinate format
Contents

- About Navi-EG
- AGL Navigation Architecture Update
  - Map Drawing Architecture
  - AGL Application Framework Binding
- **AGL Navigation API**
  - API Strategy
  - Explanation of GENIVI API
  - **AGL Extended**
- Future tasks
NaviAPI which NaviEG added newly is as follows.

- Guidance
  - notification of detail guidance function
- Highway(tollway) list
  - get list, update (detail information in next page)
- Turn by Turn list
  - get list, update
- Favorites
  - register, delete, get list
- Previous Destinations
  - delete, get list
Extended NaviAPI
example (Functional requirement of Highway list)
Extended NaviAPI example (Application1 of Highway list)

It is necessary to be able to change the indication point by user operation. (Application1)
Extended NaviAPI example (Application2 of Highway list)

It is necessary for indication to be replaced by a car having passed AAA. (Application2)
Extended NaviAPI
example(API defined of Highway list)

- I show a sequence of following API meeting two points
  - Independent indication is possible by each application
  - Update of the information is possible by a car running

![Diagram showing API interactions]

- Application1
  - start
  - getInfo(n, n+2)
  - getInfo(n+1, n+3)
  - getInfo(n+2, n+4)
- Application2
  - start
  - update(current:n)
  - getInfo(n, n+2)
- hwylist
  - GuidanceChanged(hwy:active)
  - update(current:n)
  - update(current:n+1)
- Guidance
  - passed AAA by running
  - n : Consecutive numbers from the route start point

user operation
Contents

- About Navi-EG
- AGL Navigation Architecture Update
  - Map Drawing Architecture
  - AGL Application Framework Binding
- AGL Navigation API
  - API Strategy
  - Explanation of GENIVI API
  - AGL Extended

Future tasks
## RoadMap

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NaviAPI spec 1.0</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>✤</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>NaviAPI Binder implementation</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>DeviceAPI spec 1.0</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>✤</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Hiway list</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>sample implementation</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>turn bu turn list</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>sample implementation</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Favorite</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>sample implementation</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>PreviosDestination</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>sample implementation</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>free Map data investigation</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

© AISIN AW CO., LTD. All Rights Reserved.
Future discussion

- Navigation API divide to privileged Application API and unprivileged Application API.
  - The route guide has to run only one route in system. Because when two navigation applications run own route guide, driver is confused.
  - We defined to the guide route setting API need to privilege. It need or not.

- OSS navigation maintenance
  - Currently, GPS-navi have only two area map database. It does not include Las Vegas map.
    - It is not an obstacle for Navi-EG development. But, it is a problem for CES demonstration.
    - NDS format is not open.

- API documentation
  - Json format does not have standard documentation. (doxygen, Javadoc, etc..)
  - We have to chose documentation format.

- AGL Navigation device API
For CES2018 official demo

- Our plan to CES2018 official demo
  - Migrating to new window manager and homescreen.
  - Migrating to new audio manager
    - We choice GENIVI based audio manager.
  - Replace POI apps API call from d-bus to binder.
    - GPS navi export GENIVI API using d-bus. (It is not change.)
    - We implement to Navi API binder.
Thank you.