

2

GENIVI Alliance

- 3 GENIVI Document CS00051
- 4 EnhancedPositionService
- 5 Component Specification
- 6 Accepted Version 4.0.0
- 7 **11-Feb-2016**
- 8 Sponsored by:
- 9 GENIVI Alliance
- 10 Abstract:
- 11 This document provides the Component Specification for the EnhancedPositionService
- 12 **Keywords:**
- 13 GENIVI, EnhancedPositionService, GPS, GNSS, Sensors, Dead-Reckoning.
- 14 License:
- 15 This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

1 2 3 4	Copyright © 2014, BMW Car IT GmbH, Continental Automotive GmbH, PCA Peugeot Citroën, XS Embedded GmbH, TomTom International B.V., Alpine Electronics R&D Europe GmbH, AISIN AW CO.LTD. All rights reserved.		
5 6 7 8 9	The information within this document is the property of the copyright holders and its use and disclosure are restricted. Elements of GENIVI Alliance specifications may be subject to third party intellectual property rights, including without limitation, patent, copyright or trademark rights (and such third parties may or may not be members of GENIVI Alliance). GENIVI Alliance and the copyright holders are not responsible and shall no be held responsible in any manner for identifying, failing to identify, or for securing proper access to or use of, any or all such third party intellectual property rights.		
11 12 13	GENIVI and the GENIVI Logo are trademarks of GENIVI Alliance in the U.S. and/or other countries. Other company, brand and product names referred to in this document may be trademarks that are claimed as the property of their respective owners.		
14	This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.		
15	The full license text is available at http://creativecommons.org/licenses/by-sa/4.0		
16	The above notice and this paragraph must be included on all copies of this document that are made.		
17 18 19	GENIVI Alliance 2400 Camino Ramon, Suite 375 San Ramon, CA 94583, USA		

1 Revision History

2 The following table shows the revision history for this document.

Document Revision History

Date	Version	Author	Description
10-Dec-2014	3.0.0- alpha	Marco Residori, XS Embedded (now part of Mentor Graphics)	Updated API documentation and sequence diagrams. This is the first version of this document that uses the new GENIVI component specification template. Improvements after EGLBS review
19-Jan-2015	3.0.0- alpha	Helmut Schmidt Continental Automotive GmbH	Update text according remaining review comments
21.Jan-2015	3.0.0	Marco Residori, XS Embedded (now part of Mentor Graphics)	Changed status to "Accepted"
16-Dec-2015	4.0.0- alpha	Marco Residori, Mentor Graphics	Updated API documentation in preparation to Release 4.0.0
25-Jan-2016	4.0.0	Marco Residori, Mentor Graphics	Release 4.0.0
11-Feb-2016	4.0.0	Marco Residori, Mentor Graphics	Updated document ID (26 → 51) as requested by SAT

4

Table of Contents

2	1	Introduction	1
3		1.1 System Overview	1
4		1.2 Component Overview	
5		1.3 Document Overview	
6	2	References	2
7	3	Glossary	3
0	4	Requirements	4
8	4	•	
9	5	Constraints and Assumptions	5
10	6	Architecture	6
11		6.1 Architecture Overview	6
12		6.1.1 Component Dependencies	
13		6.1.2 Component Traceability	
14		6.2 EnhancedPositionService	
15		6.2.1 Responsibility and Features	
16		6.2.2 Provided Interfaces	
17		6.2.3 Required Interfaces	
18		6.3 GNSSService	
19		6.3.1 Responsibility and Features	
20		6.3.2 Provided Interfaces	
21		6.3.3 Required Interfaces	
22		6.4 SensorsService	
23		6.4.1 Responsibility and Features	
24		6.4.2 Provided Interfaces	
25		6.4.3 Required Interfaces	
26	7	Collaboration	12
27		7.1 Get Enhanced Position	12
28		7.1.1 MapViewer retrieves enhanced position	12
29		7.1.2 NavigationCore retrieves enhanced position	
30		7.2 Get Rotation Rate	
31		7.2.1 LBS Application retrieves rotation rate	
32		7.3 Get Satellite Details	
33		7.3.1 Navigation Application retrieves satellite information	
34		7.4 Set Navigation System	
35		7.4.1 Navigation Application sets navigation system	16
36	8	Implementation	
37		8.1 Available Implementation details	17
38		8.2 Usage examples	17
39		8.3 Test Plan	17
40	9	Interfaces	
41		9.1 D-Bus	
42		9.2 Git Repository	
43		9.3 Naming Conventions	
44		9.4 Data Types Convention	
45		9.5 Errors	19
46			

1 1 Introduction

2 1.1 System Overview

- 3 The GENIVI Software Platform is a platform consisting of standardized middleware, application layer
- 4 interfaces and frameworks defined or adopted by the GENIVI Alliance.

1.2 Component Overview

- The EnhancedPositionService is a software component of the above mentioned GENIVI Software Platform that offers positioning information to client applications.
- 8
 9 To calculate the current vehicle position, data from a GNSS receiver (e.g. GPS data) and available vehicle
- 10 sensors (e.g. gyroscope and wheel ticks) are taken into account (dead-reckoning). In this way the
- 11 EnhancedPositionService can calculate the current position even on roads, where the GNSS signal is too weak
- 12 (e.g. in a tunnel, or in a parking garage) or too inaccurate (e.g. in a city or in a canyon).

13 1.3 Document Overview

14 This document describes the architecture and the interface of the GENIVI EnhancedPositionService.

15

1 2 References

- 2 The following standards and specifications contain provisions, which through reference in this document
- 3 constitute provisions of this specification. All the standards and specifications listed are normative references.
- 4 At the time of publication, the editions indicated were valid. All standards and specifications are subject to
- revision, and parties to agreements based on this specification are encouraged to investigate the possibility of
- 6 applying the most recent editions of the standards and specifications indicated below.
- 7 [1] "GENIVI GNSSService Component Specification" 8 http://git.projects.genivi.org/?p=lbs/positioning.git;a=tree;f=gnss-service/doc
- 9 [2] "GENIVI SensorsService Component Specification" –
 10 http://git.projects.genivi.org/?p=lbs/positioning.git;a=tree;f=sensors-service/doc
- 11 [3] GENIVI UML Model https://svn.genivi.org/uml-model/genivi/trunk

1 3 Glossary

2

Term	Definition
Global Navigation Satellite System	GNSS is a space-based satellite navigation system that provides location and time information.
Global Positioning System	GPS is a space-based GNSS maintained by the United States government.
Globalnaya navigatsionnaya sputnikovaya sistema	GLONASS is a space-based GNSS operated by the Russian Aerospace Defence Forces.
BeiDou Navigation Satellite System	BDS is a Chinese GNSS, also known as COMPASS.
Galileo	Galileo is a GNSS currently being built by the European Union (EU) and European Space Agency (ESA).
Vehicle Sensors	Vehicle sensors are sensors used for positioning calculation which are located either in the vehicle itself or directly in the unit where the EnhancedPositionService is deployed. Examples are Gyroscopes, Accelerometers, wheel tick or vehicle speed sensors.
Dead Reckoning	In strict sense: A technique that calculates the current position of a vehicle by integrating the relative changes in heading and distance over time since leaving a known starting point. The starting point can be determined e.g. from a GNSS system and the heading and distance changes can be determined from the vehicle sensors. In a more common sense: The fusion of GNSS and vehicle sensor data to calculate improved position and velocity. I.e. even when a GNSS fix is available.
	Global Navigation Satellite System Global Positioning System Globalnaya navigatsionnaya sputnikovaya sistema BeiDou Navigation Satellite System Galileo Vehicle Sensors

Table 1 – Acronym and Term Definitions

Requirements 1 4

- 2 The requirements related to the EnhancedPositionService are located in the GENIVI UML model (see [3]) in the
- 3 package GENIVI Model/LogicalView/SW Platform requirements/Location Based Services/Positioning.

5 Constraints and Assumptions

- 2 This is a handwritten chapter that summarizes the constraints and assumptions done in the project for the
- 3 component.

6 **Architecture** 1

3

4

5

6

7

8

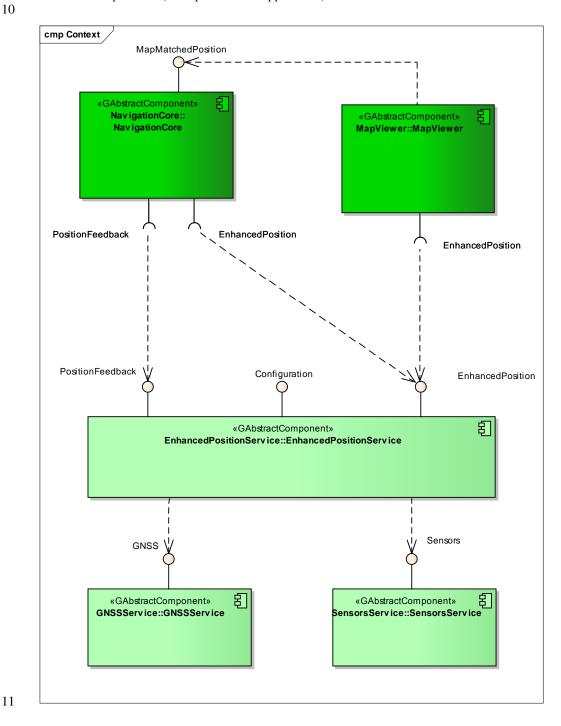
9

2 The information in this chapter is provided only for information purpose; this is not a normative part.

6.1 **Architecture Overview**

The following component diagram shows how the EnhancedPositionService interacts with other GENIVI components:

- GNSSService (C library)
- SensorsService (C library)
- NavigationCore (example of client application)
- MapViewer (example of client application)



6.1.1 Component Dependencies

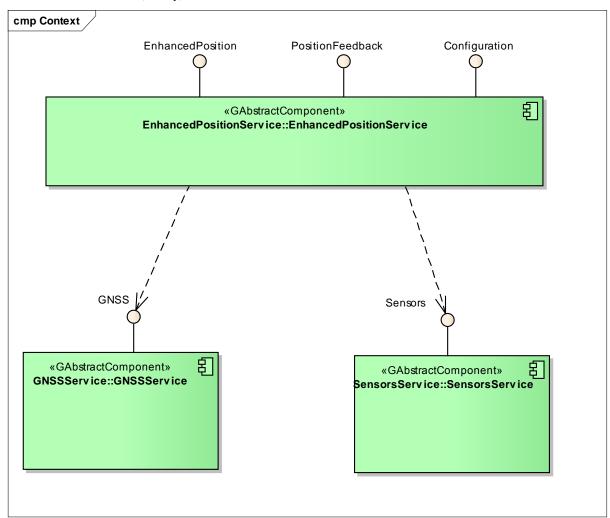
- 2 The EnhancedPositionService depends on the following GENIVI components:
 - GNSSService (library)

1

3

4

SensorsService (library)

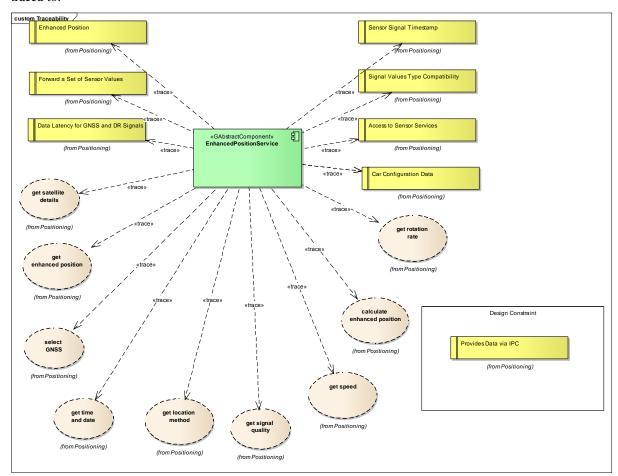


2

6.1.2 Component Traceability

3 The following diagrams shows to which requirements and use cases realizations the EnhancedPositionService is

4 traced to:



3

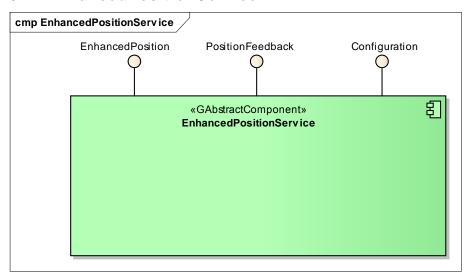
4

6 7

11

18

6.2 EnhancedPositionService



6.2.1 Responsibility and Features

- 5 The EnhancedPositionService is a software component that offers positioning information to client applications.
 - To calculate the current vehicle position, data from a GNSS receiver (e.g. GPS data) and available vehicle sensors (e.g. gyroscope and wheel ticks) are taken into account (dead-reckoning). In this way the
- 8 sensors (e.g. gyroscope and wheel ticks) are taken into account (dead-reckoning). In this way the EnhancedPositionService can calculate the current position even on roads, where the GNSS signal is too weak
- 10 (e.g. in a tunnel, or in a parking garage).

12 The result of the map matching can be provided as feedback to this module by the NavigationCore component.

- This component is the main client of the GNSSService and of the SensorsService.
- 14 The EnhancedPositionService will be typically implemented as a multi-client daemon with a D-Bus interface.

15 6.2.2 Provided Interfaces

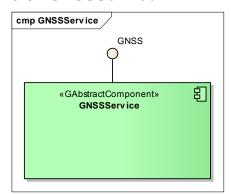
- **EnhancedPosition**: This interface provides a 'filtered' position that takes into account the value coming from the vehicle sensors (dead-reckoning).
- **PositionFeedback**: This interface offers methods that allows the NavigationCore to provide a position feedback to the EnhancedPositionService. The component that implements the Position-Feedback interface requires the data provided by a 'map matcher' (typically the NavigationCore component). The PositionFeedback is an added improvement which does not negatively affect systems that don't support maps or have a mapmetabing feeture.
- 23 matching feature.24
- Configuration: This interface allows a client application to manage configuration parameters, like the GNSS type.

27 6.2.3 Required Interfaces

- GNSS: This interface abstracts the access to a GNSS device. Please see [1].
- **Sensors**: This interface abstracts the access to vehicle sensors. Please see [2].

2

6.3 GNSSService



3

4 6.3.1 Responsibility and Features

- 5 The GNSSService is a component that retrieves positioning data from a GNSS receiver (e.g. NMEA
- 6 sentences from a GPS receiver) and presents them to its client applications.
- 7 The GNSSService will be typically implemented as a single-client library.

8 6.3.2 Provided Interfaces

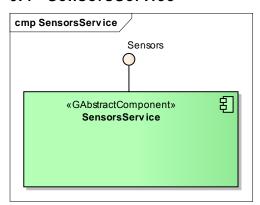
9 The interfaces provided by this component are described at [1].

10 6.3.3 Required Interfaces

11 None.

2

6.4 SensorsService



3

4 6.4.1 Responsibility and Features

- 5 The SensorsService is a component that retrieves sensor data from several vehicle sensors (e.g. gyroscope,
- 6 wheel ticks) and presents them to its client applications.
- 7 The SensorsService will be typically implemented as a single-client library.

8 6.4.2 Provided Interfaces

9 The interfaces provided by this component are described at [2].

10 6.4.3 Required Interfaces

11 None.

1 7 Collaboration

7.1 Get Enhanced Position

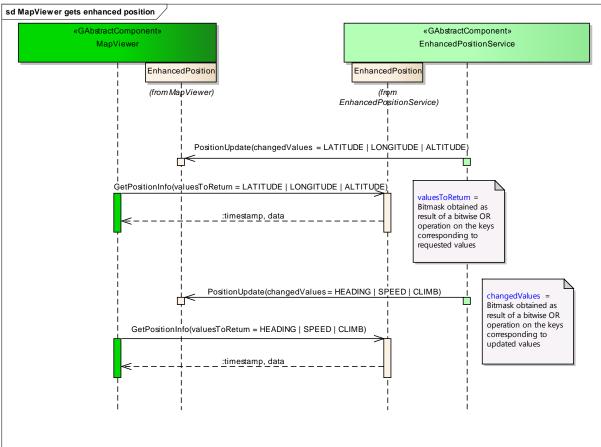
3 7.1.1 MapViewer retrieves enhanced position

4 The following sequence diagram describes how a client application can retrieve the vehicle position.

s

2

5

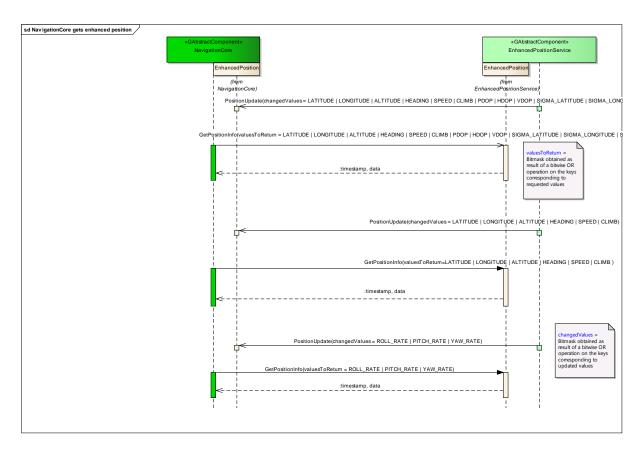


7.1.2 NavigationCore retrieves enhanced position

2 The following sequence diagram describes how a client application can retrieve the vehicle position.

3

1



1 7.2 Get Rotation Rate

7.2.1 LBS Application retrieves rotation rate

3 The following sequence diagram describes how a client application can retrieve the vehicle rotation rate.

5 6

2

7.3 Get Satellite Details

2 7.3.1 Navigation Application retrieves satellite information

3 The following sequence diagram describes how a client application can retrieve satellite information.

sd NavigationApplication

(GNamedPlaceholder»
NavigationApplication

(Inique NavigationApplication)

(Inique EnhancedPosition

(Inique EnhancedPosition)

SatelliteInfoUpdate(changedValues)

GetSatelliteInfo(satelliteInfo)

::satelliteInfo

5 6

1

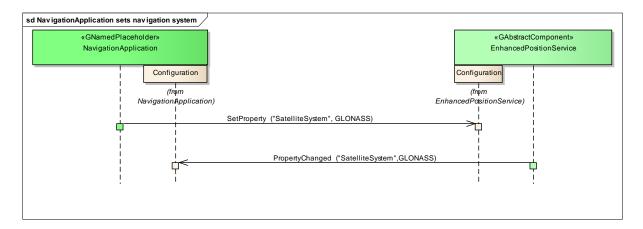
2

7.4 Set Navigation System

3 7.4.1 Navigation Application sets navigation system

4 The following sequence diagram describes how a client application can set the satellite system.

5



1 8 Implementation

2

3 8.1 Available Implementation details

- 4 A Proof of concept (PoC) of the EnhancedPositionServiceis is available at:
- 5 http://git.projects.genivi.org/?p=lbs/positioning.git;a=tree

6 8.2 Usage examples

Please see: http://git.projects.genivi.org/?p=lbs/positioning.git;a=tree;f=enhanced-position-service/test.

8 8.3 Test Plan

- 9 Please see: <a href="http://git.projects.genivi.org/?p=lbs/positioning.git;a=blob;f=enhanced-position-positio
- 10 <u>service/doc/testplan.txt</u>

9 Interfaces

2

1

3 The following pages describe the interfaces of the EnhancedPositionService.

4

5 **9.1 D-Bus**

The EnhancedPositionService interfaces are D-Bus interfaces. They are defined using the D-Bus introspection

 $\,\,$ data format, which is nothing but an IDL expressed in XML format. $\,$ 8 $\,$

9 For more information about the D-Bus data types please refer to the following website:

10 http://dbus.freedesktop.org/doc/dbus-specification.html#message-protocol-signatures

11 intp://dous.freedesktop.org/doc/dous-specification.ntm#message-protocor-signatures

For more information about the D-Bus introspection data format, please refer to the following website:

13 <u>http://dbus.freedesktop.org/doc/dbus-specification.html#introspection-format</u>
14

15

16

12

9.2 Git Repository

- 17 The EnhancedPositionService interfaces can be found in the GENIVI Git repository at:
- 18 http://git.projects.genivi.org/?p=lbs/positioning.git;a=tree;f=enhanced-position-service/api

9.3 Naming Conventions

Please see http://dbus.freedesktop.org/doc/dbus-specification.html.

21

19

Element	Description	Example
Interface File	genivi. <component domain="" in<="" name="" or="" td=""><td>org.genivi.positioning.Configuration</td></component>	org.genivi.positioning.Configuration
	lowercase character>. <interface in<="" name="" td=""><td></td></interface>	
	lowercase characters>	
Methods/Signal/Properties	Camel case naming convention	GetPositionInfo
	First letter uppercase	
Arguments	Camel case naming convention	valuesToReturn
	First letter lowercase	

22

23

24

9.4 Data Types Convention

D-bus types code are used. Please refer to the following webpage for more information:

http://dbus.freedesktop.org/doc/dbus-specification.html

25 26 27

Element	D-Bus Data Type Code	Example
Enumerators	q (uint16)	
Handles	y (uint8)	
Maps	$a\{qv\}$	Dictionary of tuples (key, value)
		The key is expressed as an enumerator

9.5 Errors

2

Error Type	Description	Example	Error	Note
7-	-	_	Documentation	
User Error	Error caused by user actions	The user tries to start route guidance, although guidance is already running	Application specific error string documented in the XML file	Can occur in final product
Hardware Error	Error related to hardware/database related problems	No map data	Application specific error string documented in the XML file	Can occur in final product
Protocol Error	Error caused by wrong sequence of commands	Wrong sequence of commands to enter destination	Standard D-Bus error string	Should not occur in final product
Bus Error	D-Bus communication error	Bus busy	Standard D-Bus error string	Can occur in final product
Programming Error	Programming Error	Invalid parameters	Standard D-Bus error string and debug messages	Should not occur in production code

4 5 6

Only application-specific errors are documented directly in the interfaces (XML files). For all other errors, standard D-Bus strings are used. These kinds of strings are not documented in the interfaces. It is implicitly assumed that every method may return a standard D-Bus error string.

7 8 9

Please see http://dbus.freedesktop.org/doc/api/html/group DBusProtocol.html.