



GENIVI NavigationCore API

Release 3.0.2
Status: Approved

30 June 2014

Accepted for release by:

This document has been accepted for the GENIVI Gemini Release by the Expert Group Location Based Services (EG-LBS)

Abstract:

This document describes the API of the NavigationCore Abstract Component.

Keywords:

NavigationAPIs, NavigationCore.

SPDX-License-Identifier: CC-BY-SA-4.0

Copyright (C) 2014, BMW Car IT GmbH, Continental Automotive GmbH, Elektrobit Automotive GmbH, Neusoft Technology Solutions GmbH, PCA Peugeot Citroën, TomTom International B.V., XS Embedded GmbH

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License

To view a copy of this license, visit

<http://creativecommons.org/licenses/by-sa/4.0/>

or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

Table of contents

1	Change History	4
2	Introduction	5
3	Terminology	6
4	Requirements	7
5	Architecture	8
5.1	Interfaces	8
5.2	Interaction with other Components	9
6	API	10
6.1	D-Bus	10
6.2	Git Repository	10
6.3	Naming Conventions	10
6.4	Data Types Convention	11
6.5	Errors	12
6.6	Sessions	13
6.7	Sequence Diagrams	15
6.7.1	navigation application creates route	15
6.7.2	navigation application starts route calculation	16
6.7.3	navigation application gets list of segments	17
6.7.4	navigation application enters destination	18
6.7.5	two clients try to change route preferences of the same route	19
6.7.6	navigation application sets route preferences	20
6.7.7	navigation application sets starting point	21
6.7.8	navigation application sets transportation means	22
6.7.9	navigation application changes waypoints order	23
6.7.10	navigation application enables voice guidance	24
6.7.11	navigation application starts a simulation	25
6.7.12	navigation application starts guidance	26
6.7.13	navigation application stops guidance	27
6.7.14	HMI requests voice instruction	28
6.7.15	navigation application creates location input session	29
6.7.16	navigation application enters location	30
6.7.17	navigation application enters location using speller	31
6.7.18	navigation application enters full address	32
6.8	Interfaces	33

1 Change History

Version	Date	Author	Change
0.1	27 Feb 2012	Marco Residori (XS Embedded)	Document Created.
0.2	19 Mar 2012	Marco Residori (XS Embedded)	Updated sequence diagrams. Updated <i>Interfaces</i> chapter.
0.3	21 Mar 2012	Marco Residori (XS Embedded)	Updated <i>Interfaces</i> chapter.
1.0	22 Mar 2012	Marco Residori (XS Embedded)	System Architecture Team (SAT) approval.
2.0 (beta)	07 Jun 2013	Marco Residori (XS Embedded)	Updated API description. API Version 2.0.
2.0	17 Jun 2013	Marco Residori (XS Embedded)	Updated API description. API fixes: GT-2691, GT-2689, GT-2651. API Version 2.0 (<i>gemini-final</i> tag)
3.0.0	21 Jan 2014	Marco Residori (XS Embedded)	Updated API description. API Version 3.0.0
3.0.1	22 May 2014	Marco Residori (XS Embedded)	Updated copyright notes.
3.0.2	30 June 2014	Marco Residori (XS Embedded)	Updated contributors list.

2 Introduction

This document describes the NavigationCore API.

3 Terminology

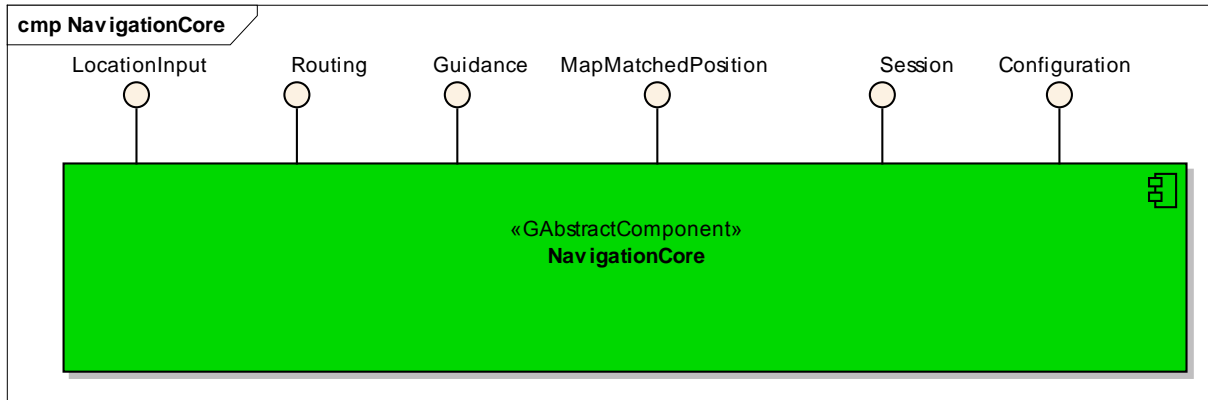
<i>Term</i>	<i>Description</i>
Link-ID	Identifier of a route segment in a database

4 Requirements

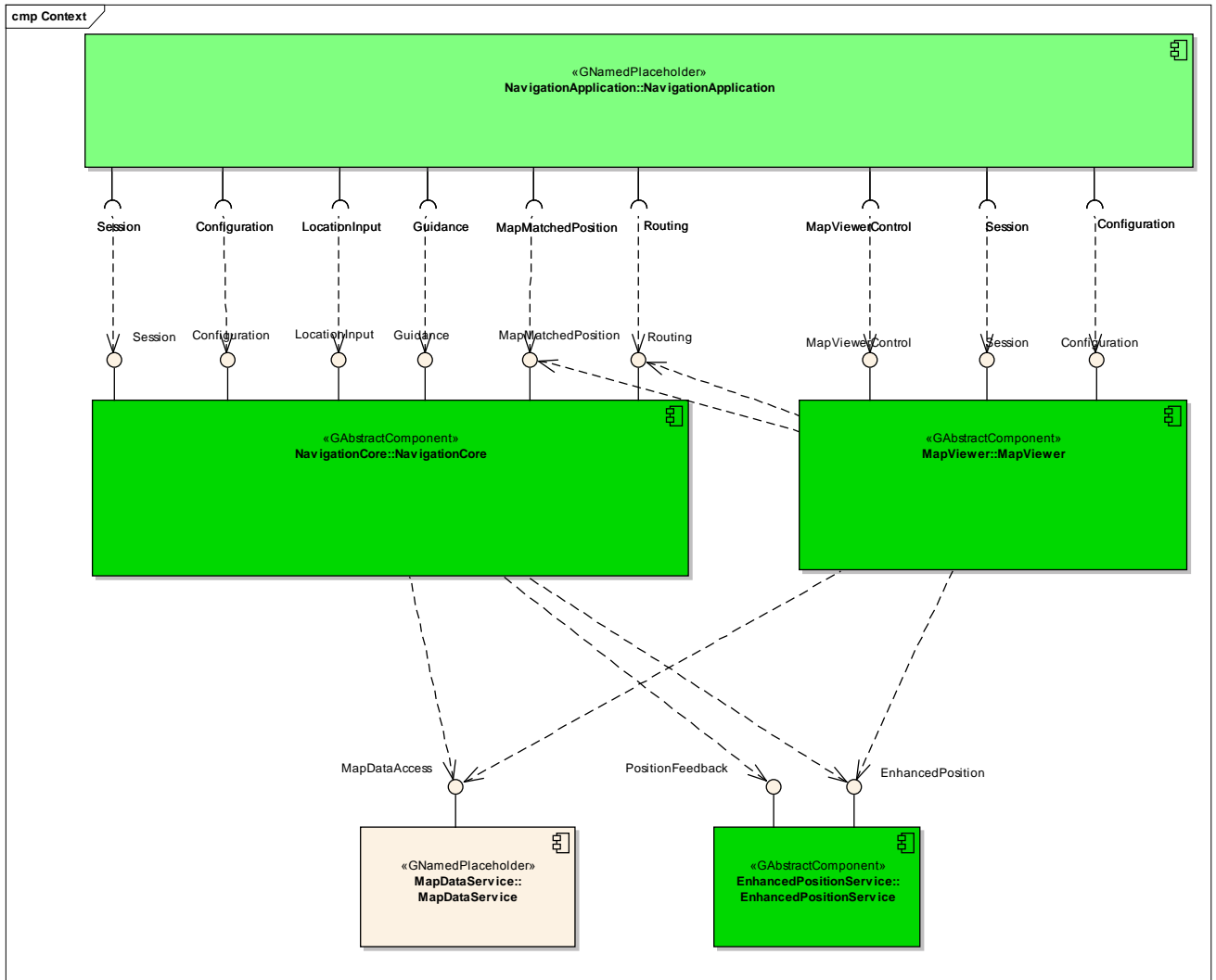
- Ease of Development
- Extensibility
- Multi-client Behavior
- Simplicity

5 Architecture

5.1 Interfaces



5.2 Interaction with other Components



6 API

6.1 D-Bus

The NavigationCore 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.

For more information about the D-Bus data types please refer to the following website:
<http://dbus.freedesktop.org/doc/dbus-specification.html#message-protocol-signatures>

For more information about the D-Bus introspection data format, please refer to the following website:
<http://dbus.freedesktop.org/doc/dbus-specification.html#introspection-format>

6.2 Git Repository

The NavigationCore interfaces can be found in the GENIVI Git repository at:
<https://git.genivi.org/git/gitweb.cgi?p=navigation;a=tree;f=NavigationCore/api>

6.3 Naming Conventions

Element	Description	Example
Interface File	genivi.<component name in lowercase character>.<interface name in lowercase characters>	genivi.navigationcore.routing.xml
Methods/Signal/Properties	Camel case naming convention First letter uppercase	CalculateRoute
Arguments	Camel case naming convention First letter lowercase	routeHandle

6.4 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>

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

6.5 Errors

Error Type	Description	Example	Error Documentation	Note
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

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.

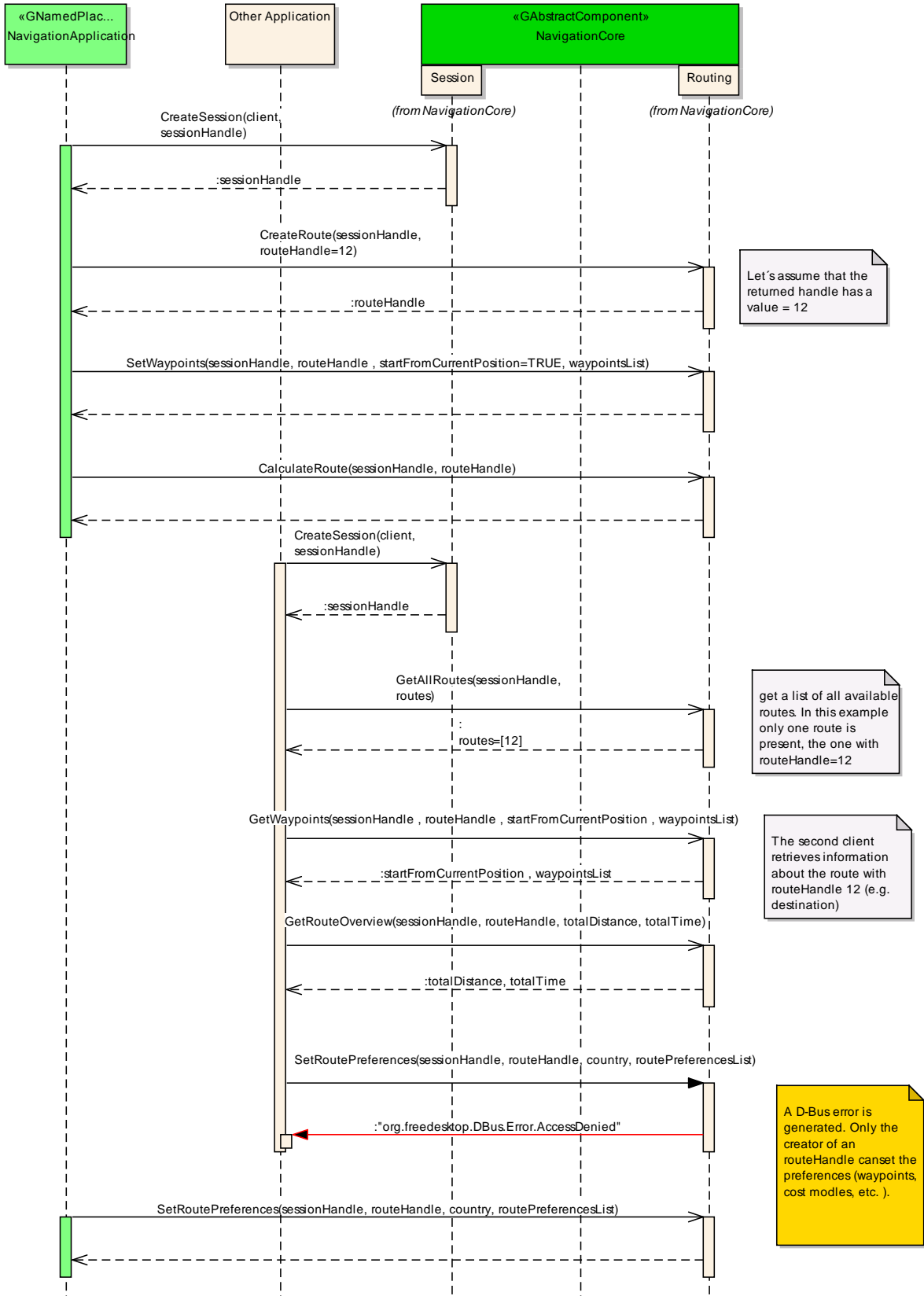
6.6 Sessions

A session-id identifies a requester. In a multi-client context only the requester that created an instance may be allowed to execute operations on that instance.

Other requesters may operate on the same handle (e.g. route handle, location input handle) with limited rights. For example, only the creator of a route handle can start/stop a route calculation on that handle. Other clients may simply be allowed to retrieve the total distance to the destination.

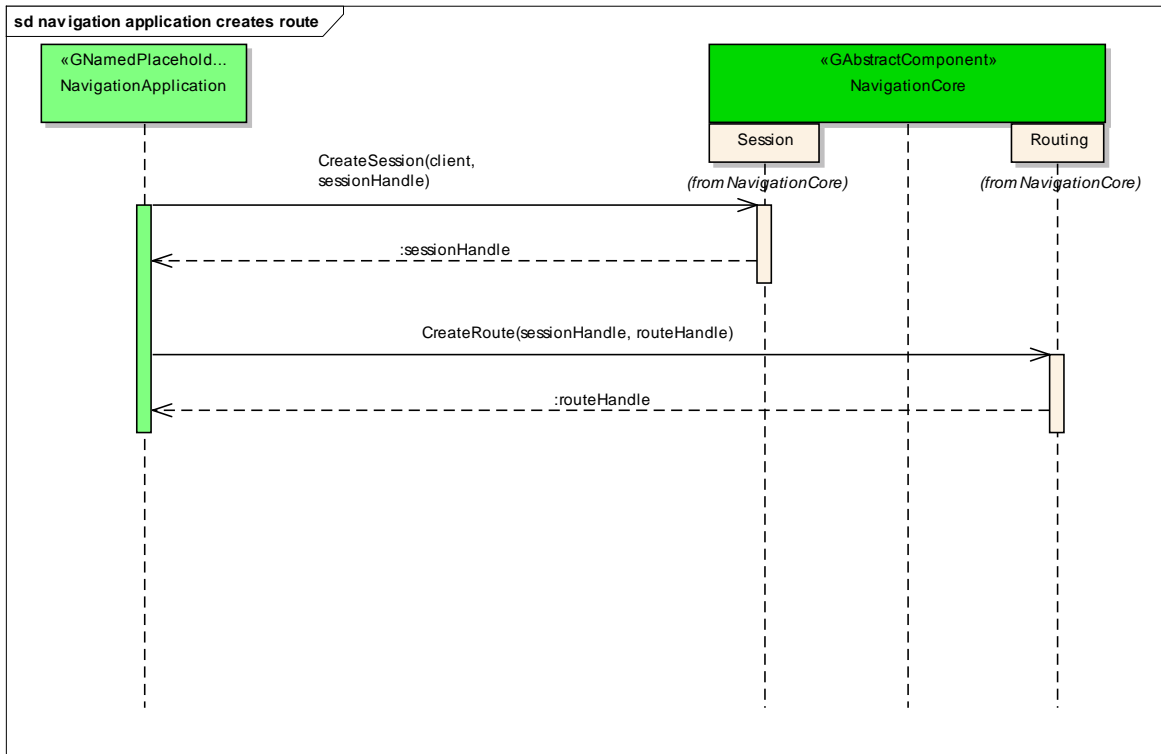
The following diagram shows an example of how session handles can be used:

sd two clients try to change route preferences of the same route

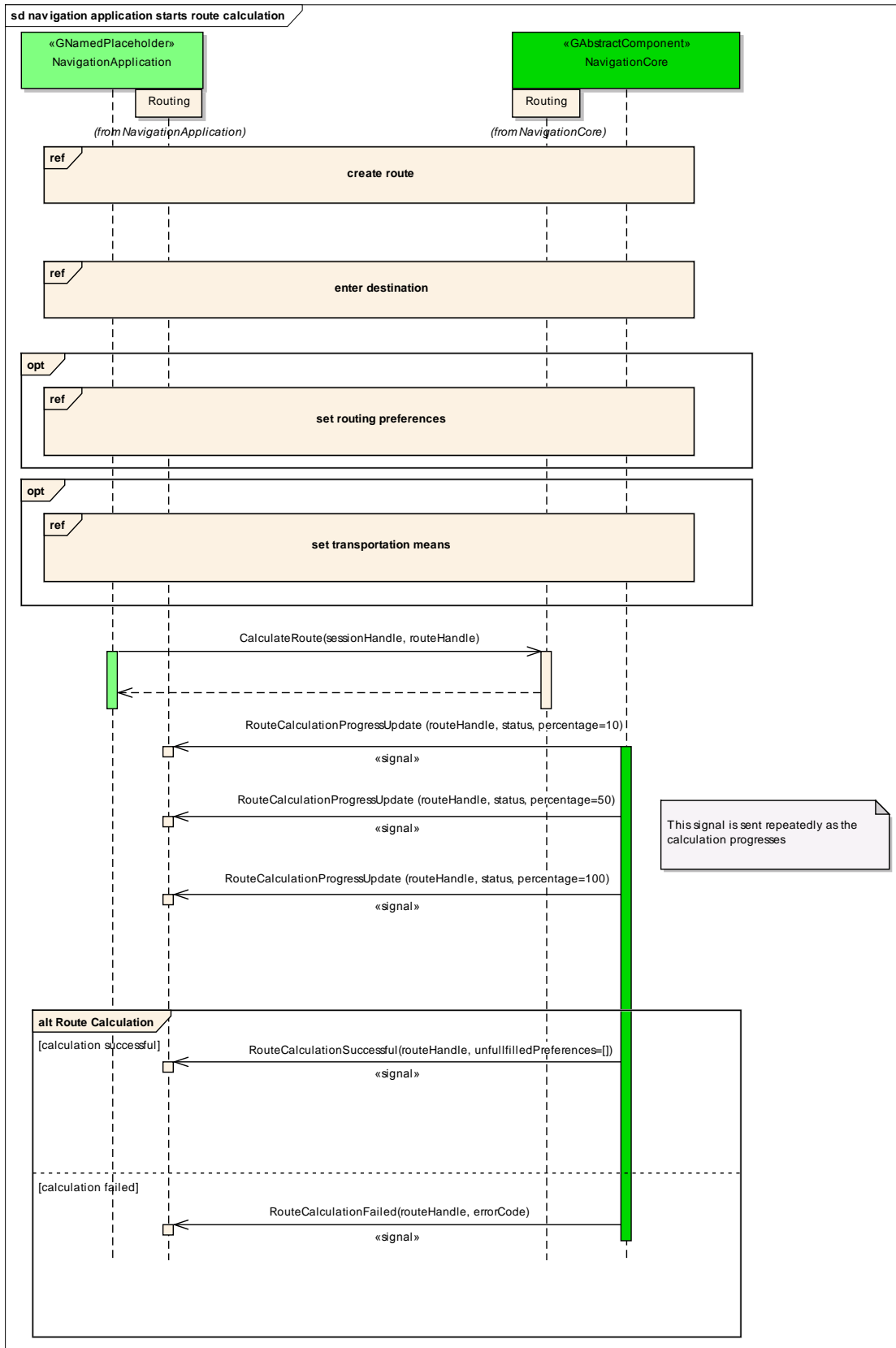


6.7 Sequence Diagrams

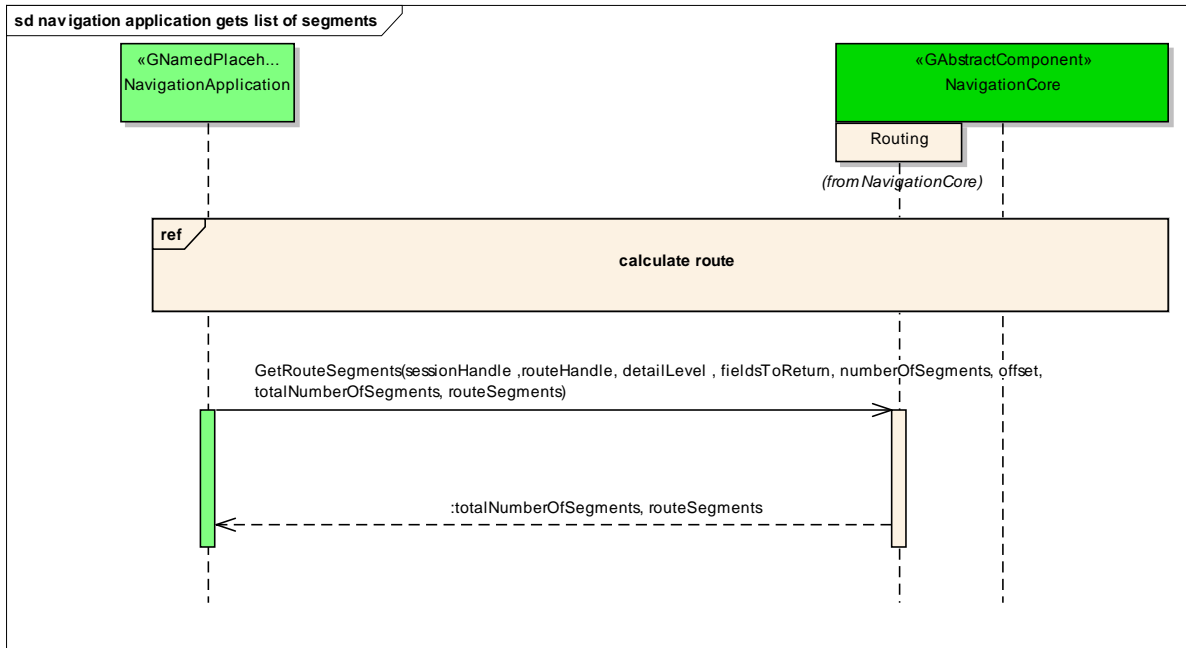
6.7.1 navigation application creates route



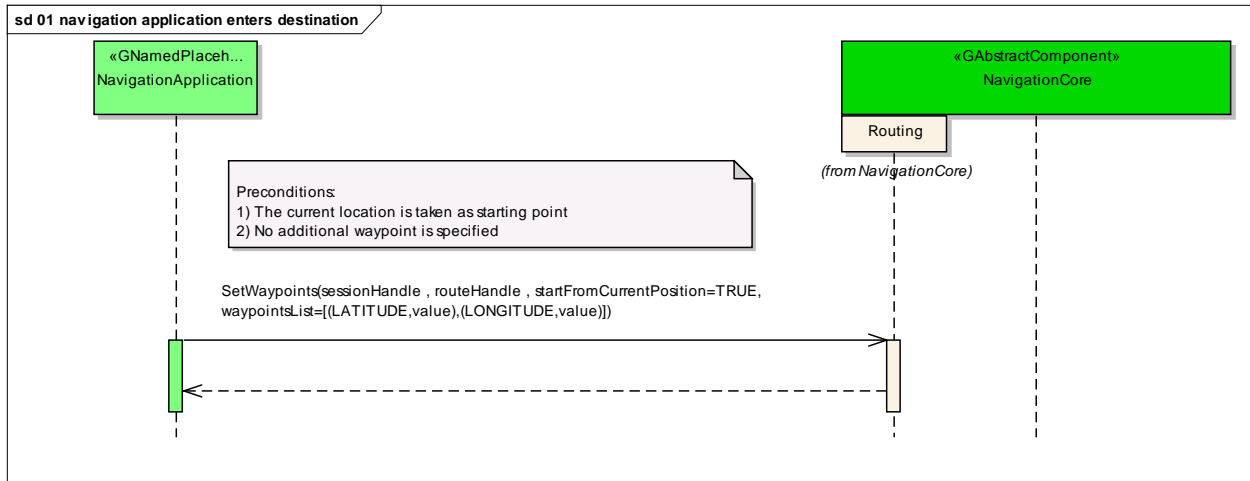
6.7.2 navigation application starts route calculation



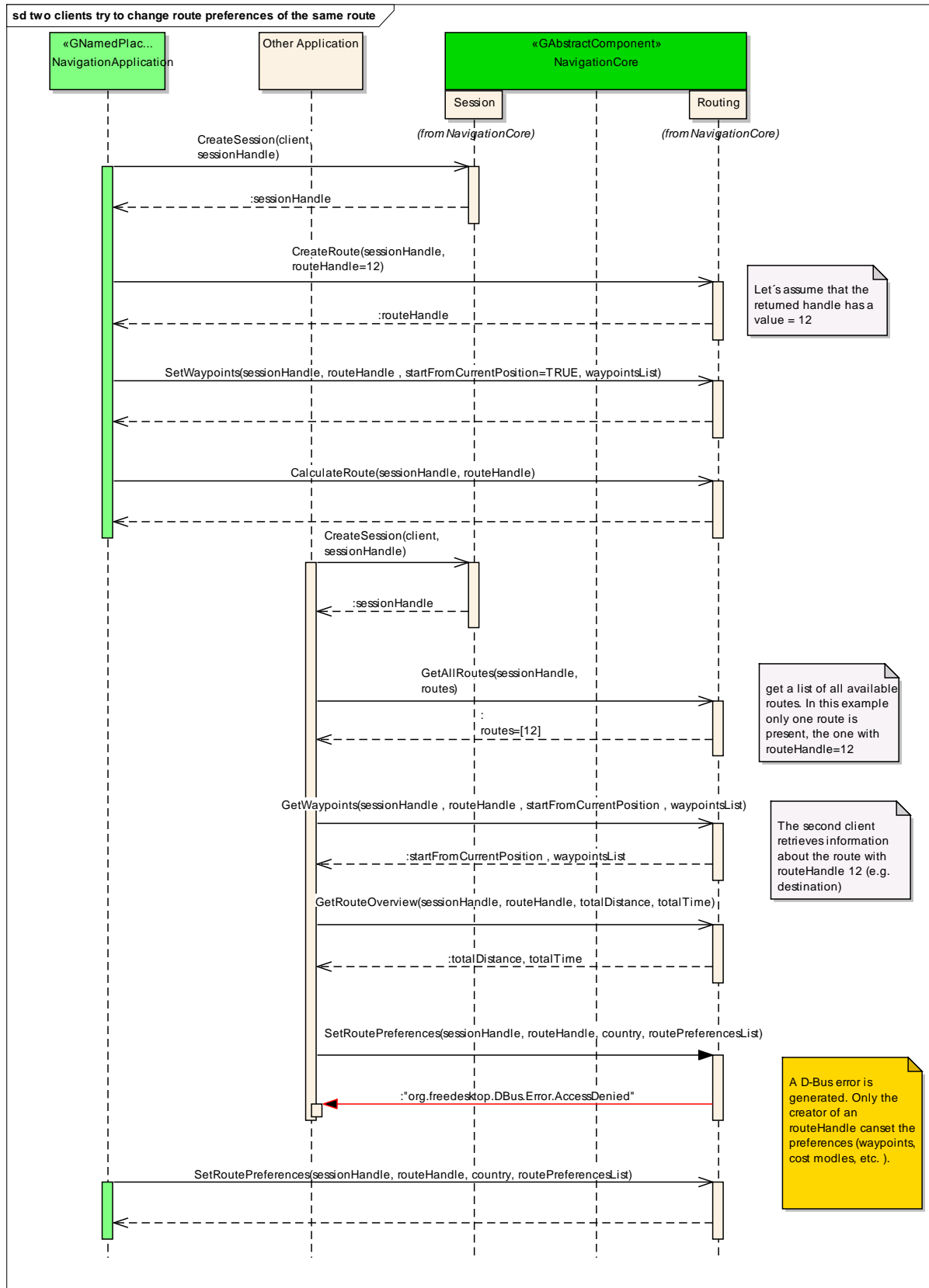
6.7.3 navigation application gets list of segments



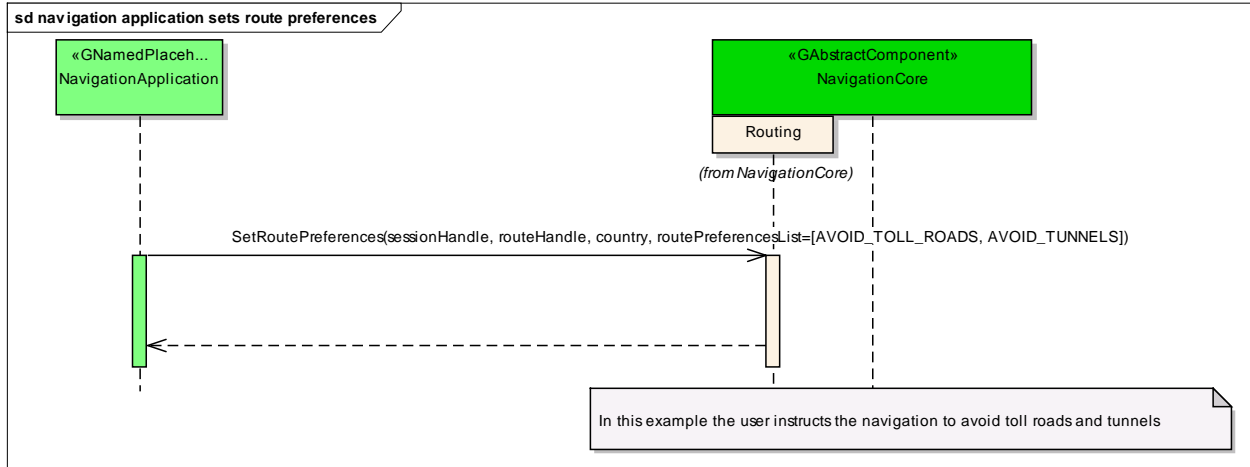
6.7.4 navigation application enters destination



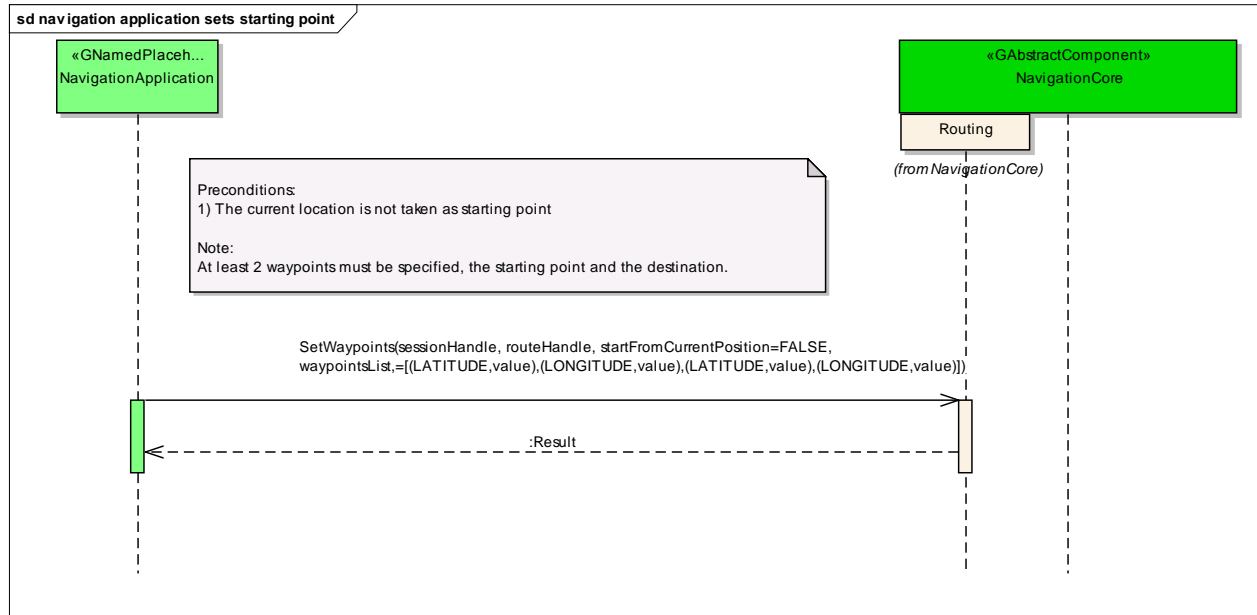
6.7.5 two clients try to change route preferences of the same route



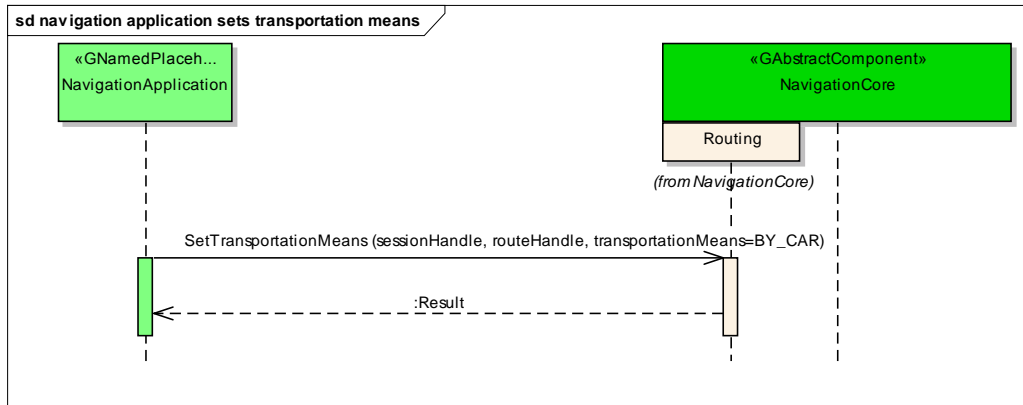
6.7.6 navigation application sets route preferences



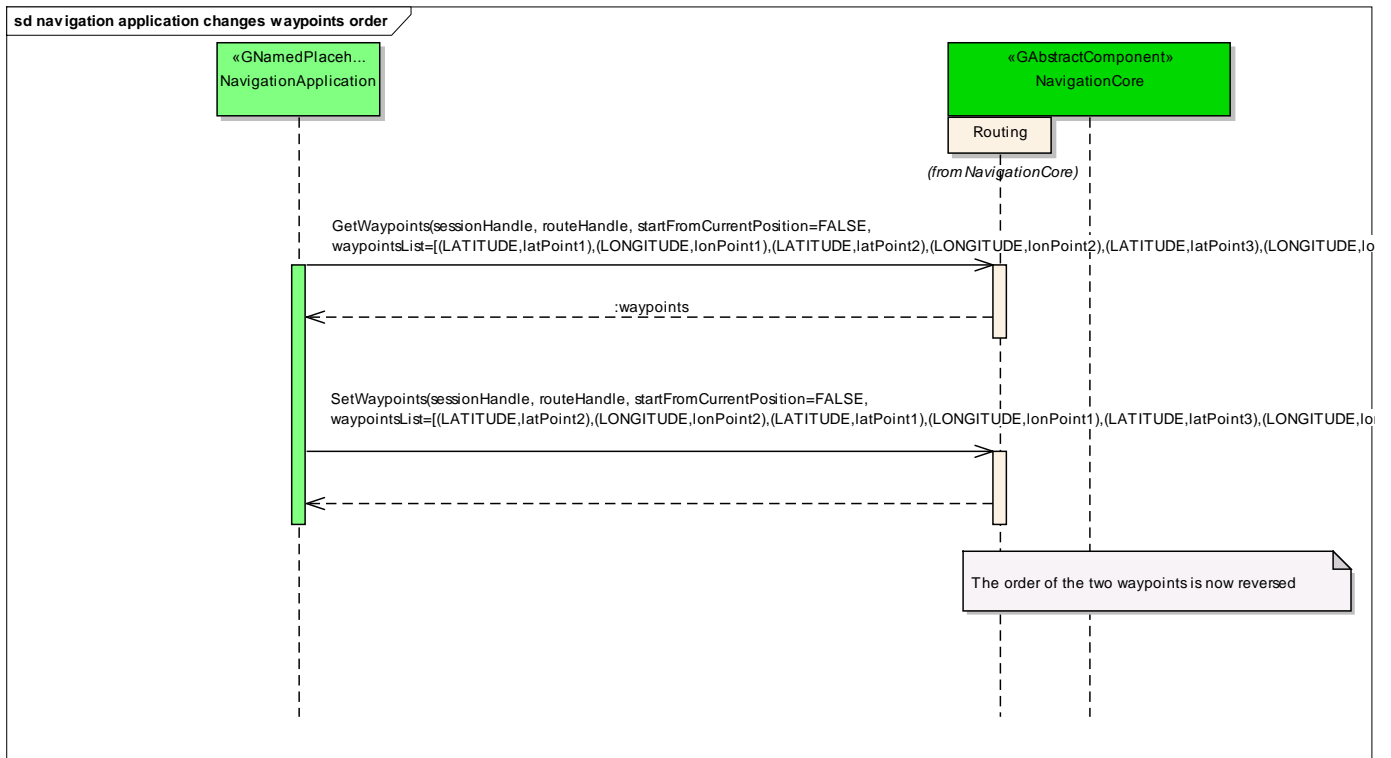
6.7.7 navigation application sets starting point



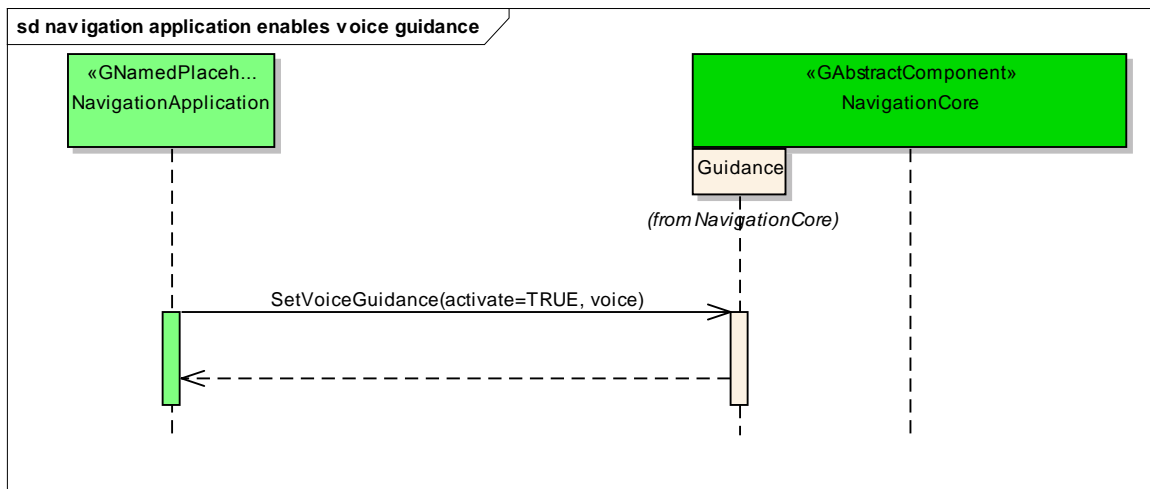
6.7.8 navigation application sets transportation means



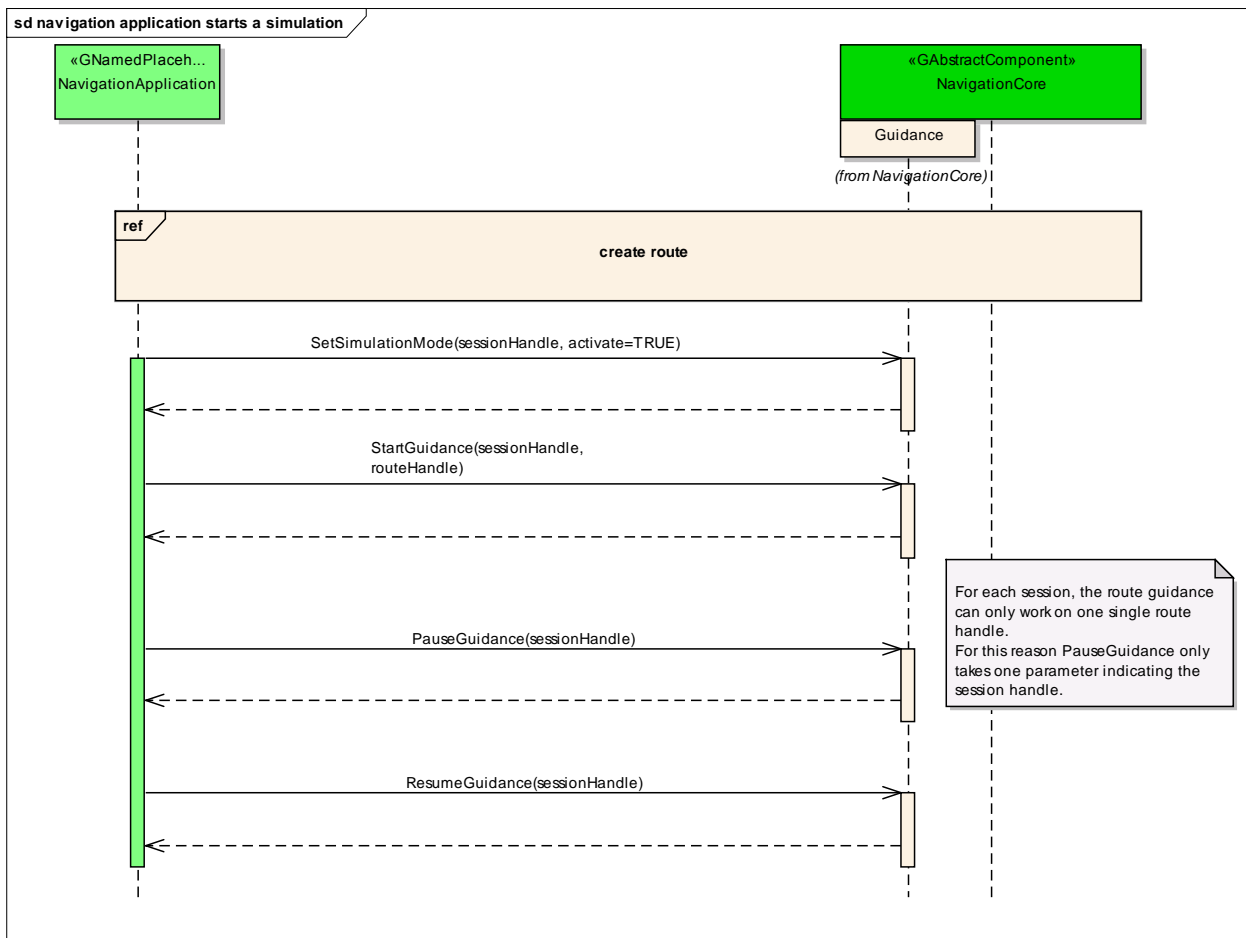
6.7.9 navigation application changes waypoints order



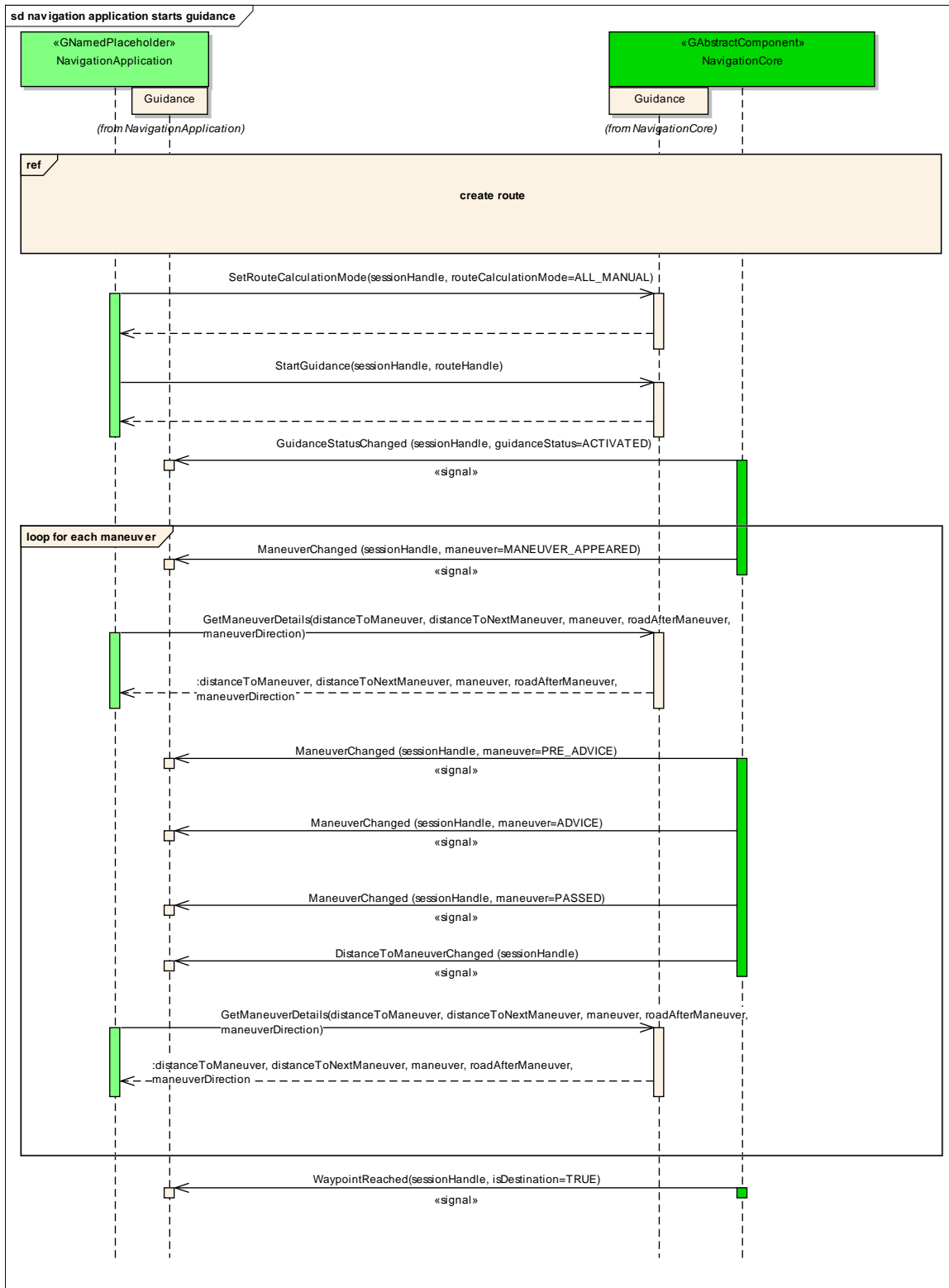
6.7.10 navigation application enables voice guidance



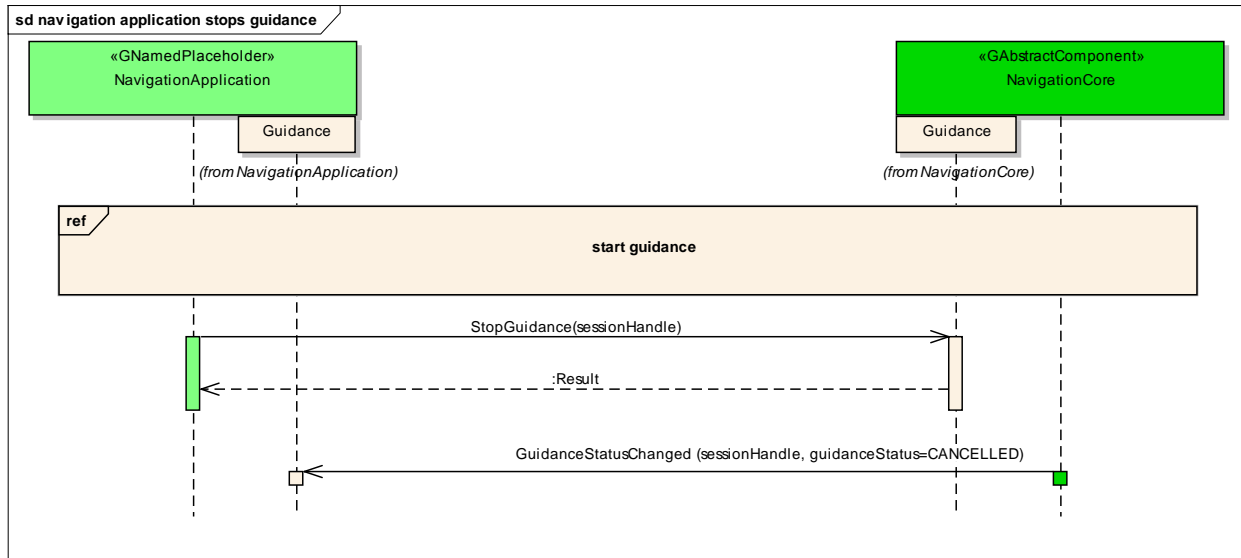
6.7.11 navigation application starts a simulation



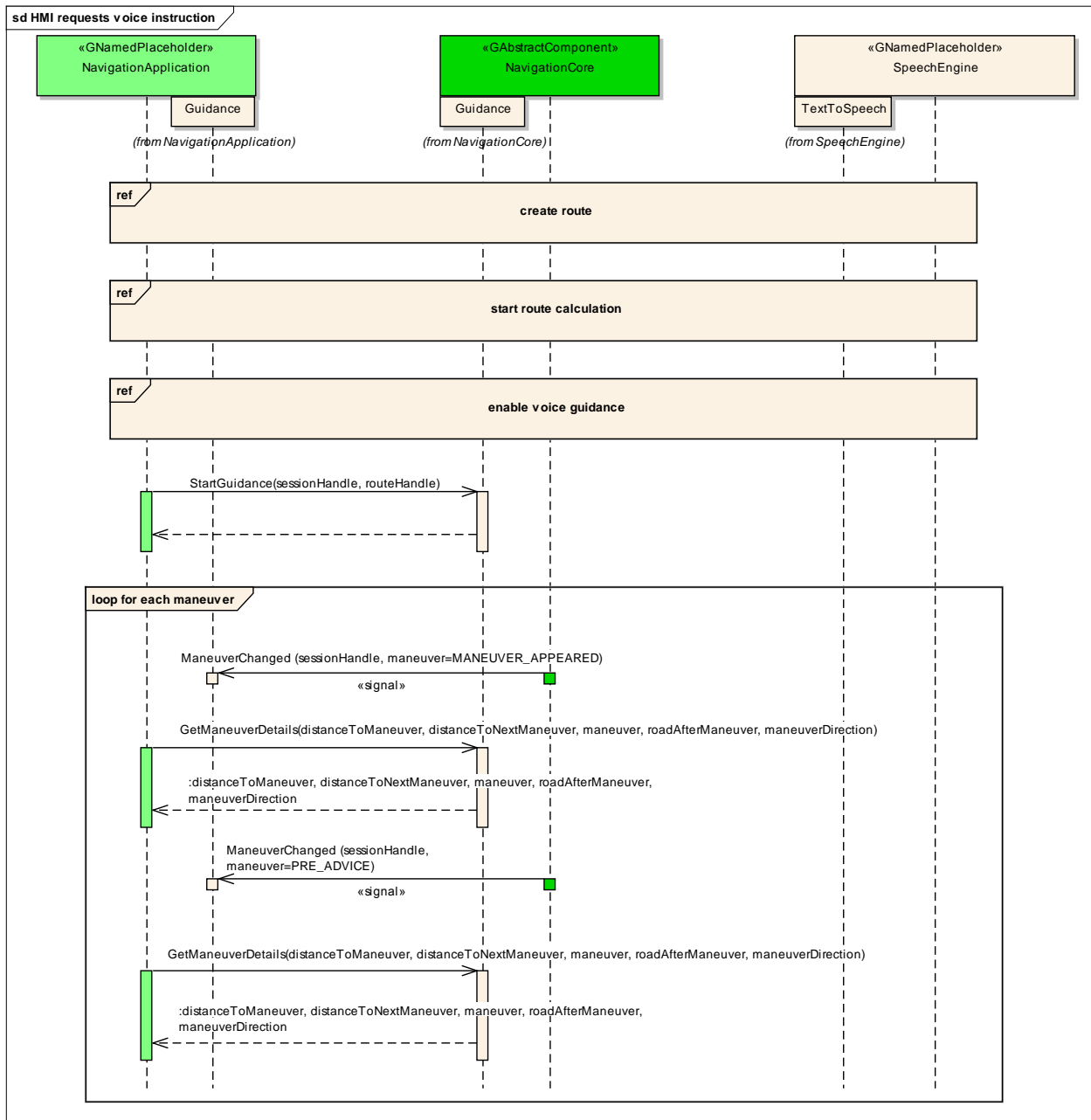
6.7.12 navigation application starts guidance



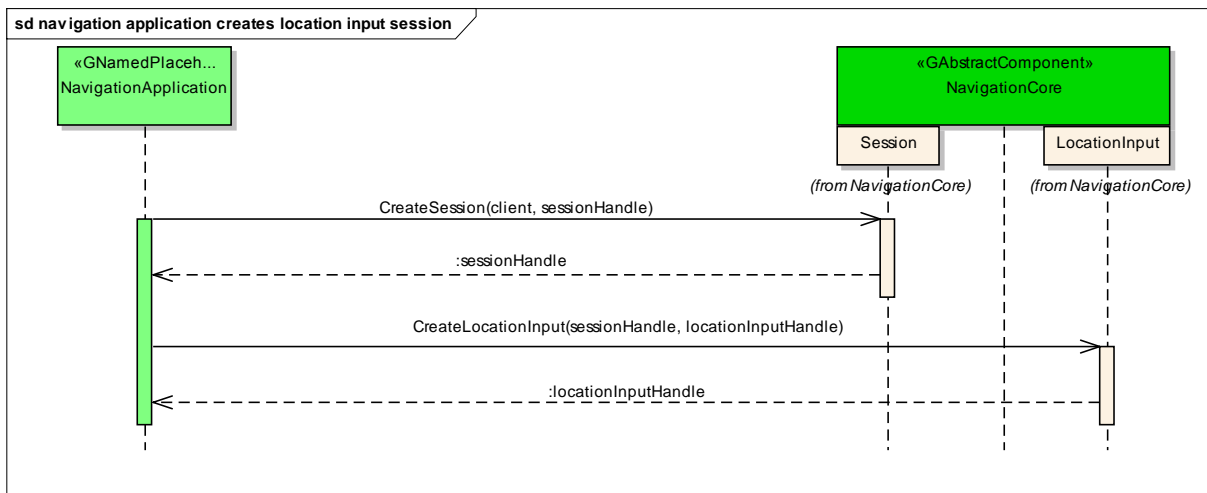
6.7.13 navigation application stops guidance



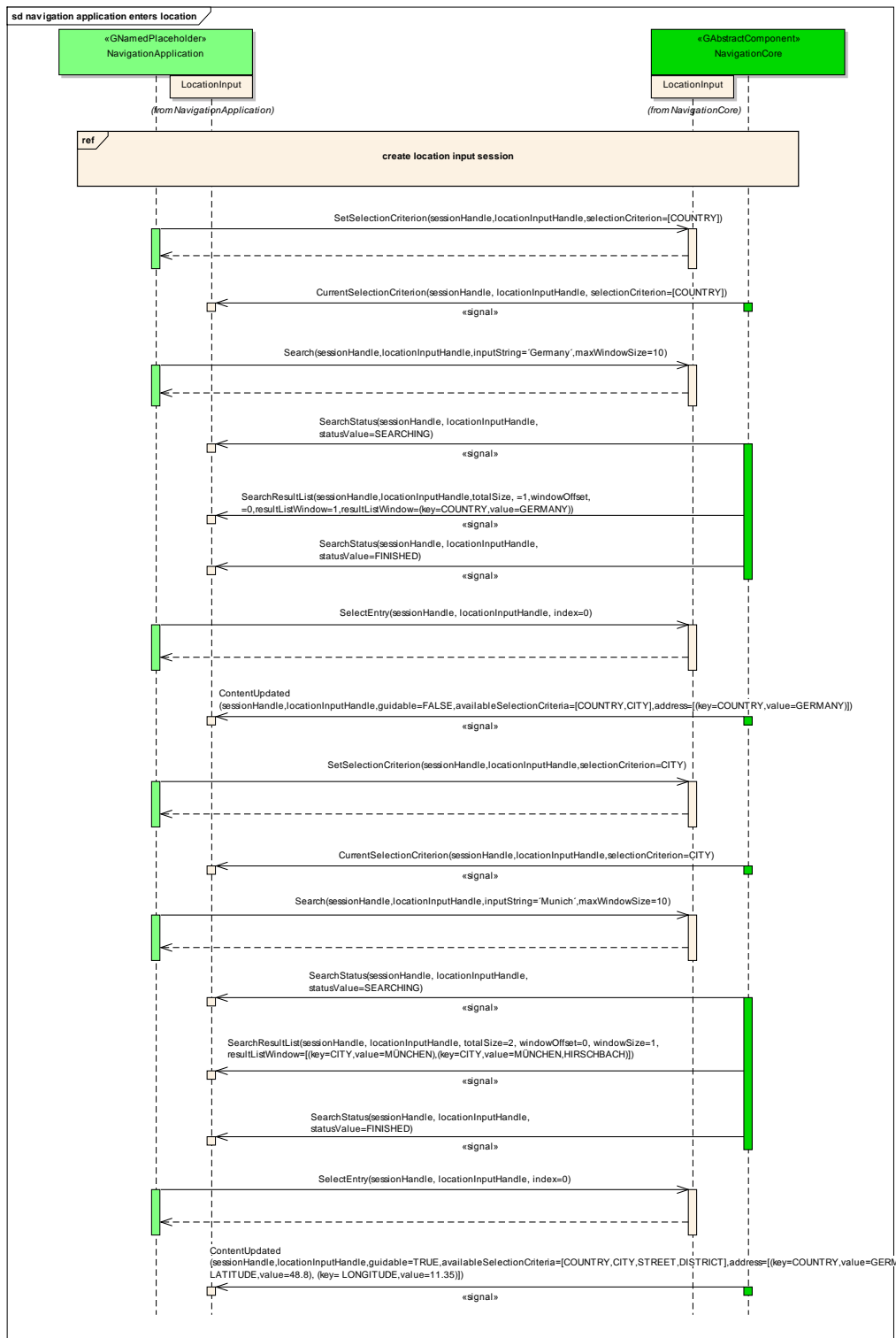
6.7.14 HMI requests voice instruction



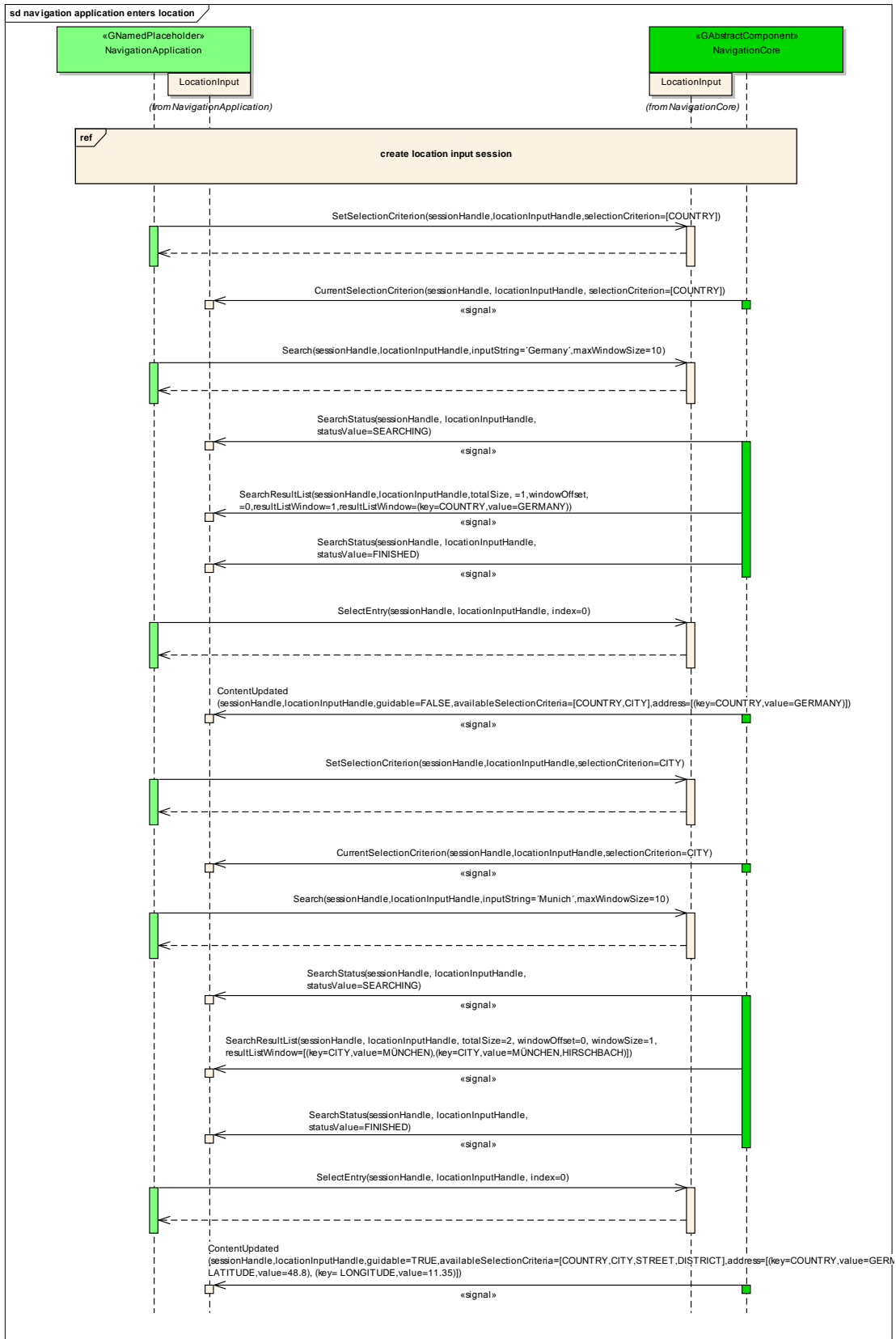
6.7.15 navigation application creates location input session



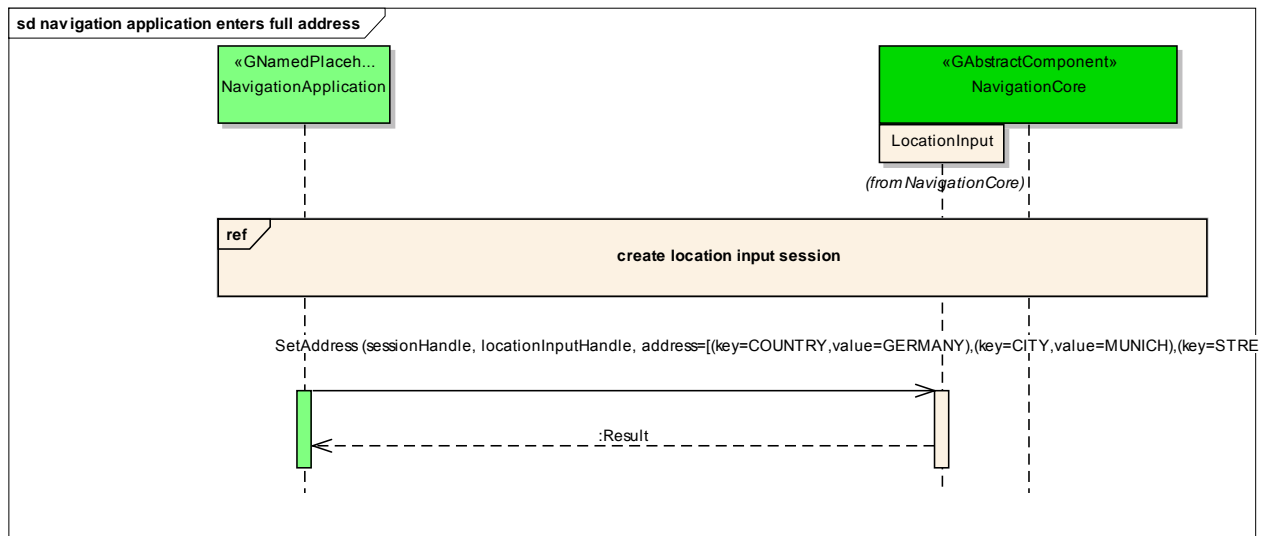
6.7.16 navigation application enters location



6.7.17 navigation application enters location using speller



6.7.18 navigation application enters full address



6.8 Interfaces