Final draft **ETSI ES 202 196-5** V1.1.1 (2003-06)

ETSI Standard

Open Service Access (OSA);
Application Programming Interface (API);
Test Suite Structure and Test Purposes (TSS&TP);
Part 5: User interaction SCF

Reference

DES/SPAN-120088-5

Keywords

API, OSA, TSS&TP

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or
perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF).
In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive
within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, send your comment to:
editor@etsi.org

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This ETSI Standard (ES) has been produced by ETSI Technical Committee Services and Protocols for Advanced Networks (SPAN), and is now submitted for the ETSI standards Membership Approval Procedure.

The present document is part 5 of a multi-part deliverable. Full details of the entire series can be found in part 1 [].

To evaluate conformance of a particular implementation, it is necessary to have a set of test purposes to evaluate the dynamic behaviour of the Implementation Under Test (IUT). The specification containing those test purposes is called a Test Suite Structure and Test Purposes (TSS&TP) specification.

1 Scope

The present document provides the Test Suite Structure and Test Purposes (TSS&TP) specification for the User Interaction SCF of the Application Programming Interface (API) for Open Service Access (OSA) defined in ES 201 915-5 [1] in compliance with the relevant requirements, and in accordance with the relevant guidance given in ISO/IEC 9646-2 [4] and ETS 300 406 [5].

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

[1]   ETSI ES 201 915-5: "Open Service Access (OSA); Application Programming Interface (API); Part 5: User Interaction SCF".

[2]   ETSI ES 202 170: "Open Service Access (OSA); Application Programming Interface (API); Implementation Conformance Statement (ICS) proforma specification for Framework and SCFs".

[3]   ISO/IEC 9646-1: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 1: General concepts".

[4]   ISO/IEC 9646-2: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 2: Abstract Test Suite specification".

[5]   ETSI ETS 300 406: "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

[6]   ETSI ES 202 196-1: "Open Service Access (OSA); Application Programming Interface (API); Test Suite Structure and Test Purposes (TSS&TP); Part 1: Overview".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 201 915-5 [1], ISO/IEC 9646-1 [3] and ISO/IEC 9646-2 [4] and the following apply:

abstract test case: Refer to ISO/IEC 9646-1 [3].

Abstract Test Method (ATM): Refer to ISO/IEC 9646-1 [3].

Abstract Test Suite (ATS): Refer to ISO/IEC 9646-1 [3].

Implementation Under Test (IUT): Refer to ISO/IEC 9646-1 [3].

Lower Tester (LT): Refer to ISO/IEC 9646-1 [3].

Implementation Conformance Statement (ICS): Refer to ISO/IEC 9646-1 [3].

ICS proforma: Refer to ISO/IEC 9646-1 [3].

Implementation eXtra Information for Testing (IXIT): Refer to ISO/IEC 9646-1 [3].

IXIT proforma: Refer to ISO/IEC 9646-1 [3].

Test Purpose (TP): Refer to ISO/IEC 9646-1 [3].

3.2   Abbreviations

For the purposes of the present document, the following abbreviations apply:

ICS	Implementation Conformance Statement
IUT	Implementation Under Test
IXIT	Implementation eXtra Information for Testing
SUT	System Under Test
TP	Test Purpose
LT	Lower Tester
ATS	Abstract Test Suite
ATM	Abstract Test Method
OSA	Open Service Access
API	Application Programming Interface
TSS	Test Suite Structure

4   Test Suite Structure (TSS)

- User Interaction

    - IPUIManager interface

    - IPUI interface

    - IPUICall interface

5   Test Purposes (TP)

5.1   Introduction

For each test requirement a TP is defined.

5.1.1   TP naming convention

TPs are numbered, starting at 01, within each group. Groups are organized according to the TSS. Additional references are added to identify the actual test suite (see table 1).

Table 1: TP identifier naming convention scheme

Identifier: <suite_id>_<nnn>
   <suite_id>   = SCG name: "UI" for **U**ser **I**nteraction part of User Interaction SCF
   <nn>   = sequential number: (01-99)

5.1.2   Source of TP definition

The TPs are based on ES 201 915-5 [1].

5.1.3   Test strategy

As the base standard ES 201 915-5 [1] contains no explicit requirements for testing, the TPs were generated as a result of an analysis of the base standard and the PICS specification ES 202 170 [2].

The TPs are only based on conformance requirements related to the externally observable behaviour of the IUT and are limited to conceivable situations to which a real implementation is likely to be faced (see ETS 300 406 [5]).

5.2   TPs for the User Interaction SCF

All PICS items referred to in this clause are as specified in ES 202 170 [2] unless indicated otherwise by another numbered reference.

All parameters specified in method calls are valid unless specified.

The procedures to trigger the SCF to call methods in the application are dependant on the underlying network architecture and are out of the scope of this test specification. Those method calls are preceded by the words "Triggered action".

5.2.1   User Interaction

Pre-requisite:

- Registration of the IUT (User Interaction SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpUIManager interface through selecting that service and signing the required service agreement.

5.2.1.1   IpUIManager interface
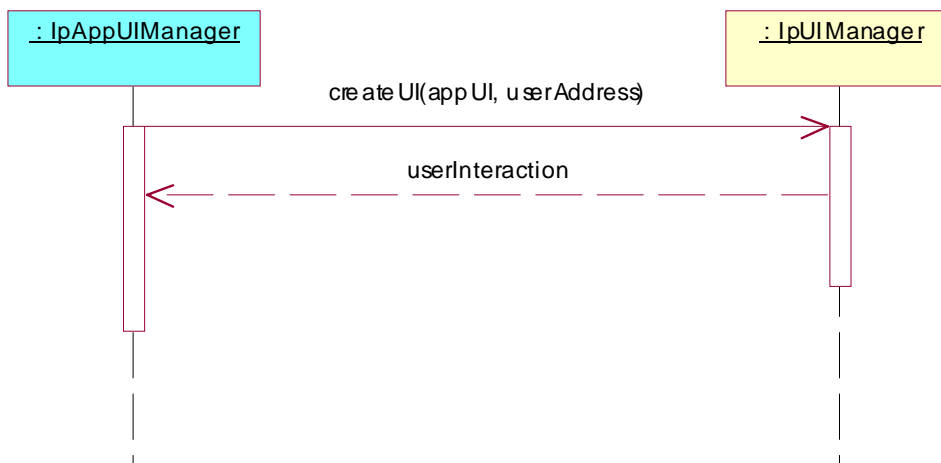
Test UI_01

Summary: **IpUIManager**, createUI method, successful

Reference: ES 201 915-5 [1], clause 8.1

Selection: createUI method supported - PICS item: [2] UIM1

Test Sequence:

1.   Method call **createUI()**
      Parameters: appUI, userAddress
      Check: valid value of TpUIIdentifier is returned

Test UI_02

Summary: **IpUIManager**, createUI method, P_INVALID_NETWORK_STATE

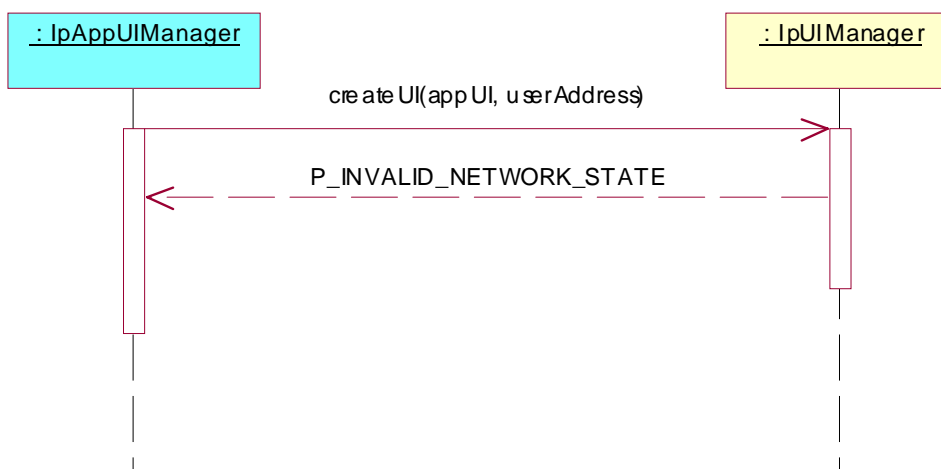Reference: ES 201 915-5 [1], clause 8.1

Selection: createUI method supported - PICS item: [2] UIM1

Condition: the underlying protocol cannot support the method call.

Test Sequence:

1.   Method call **createUI()**
     Parameters: appUI, userAddress
     Check: P_INVALID_NETWORK_STATE is returned

NOTE: Any other suitable exception is accepted.

Test UI_03

Summary: **IpUIManager**, createUI method, P_INVALID_INTERFACE_TYPE

Reference: ES 201 915-5 [1], clause 8.1

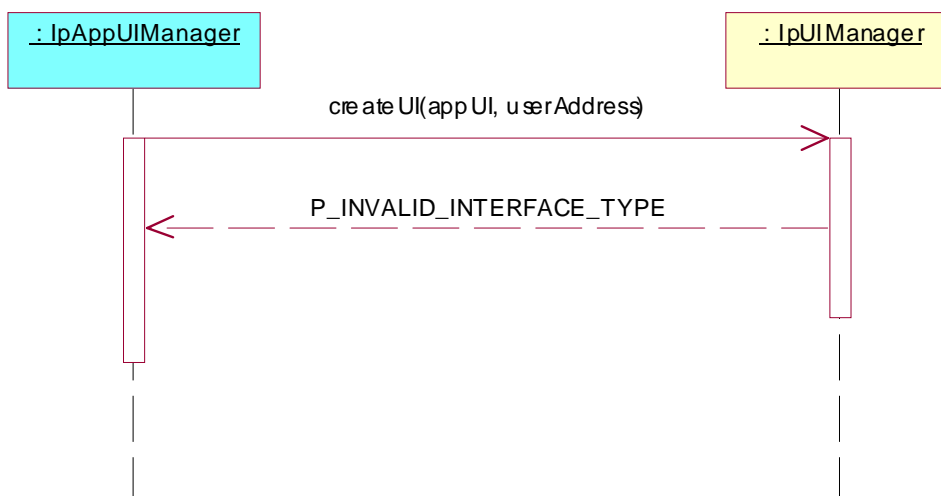Selection: createUI method supported - PICS item: [2] UIM1

Test Sequence:

1. Method call **createUI()**
    Parameters: appUI including an invalid reference type, userAddress
    Check: P_INVALID_ INTERFACE_TYPE is returned

NOTE: Any other suitable exception is accepted.

Test UI_04

Summary: **IpUIManager**, createUICall method, successful

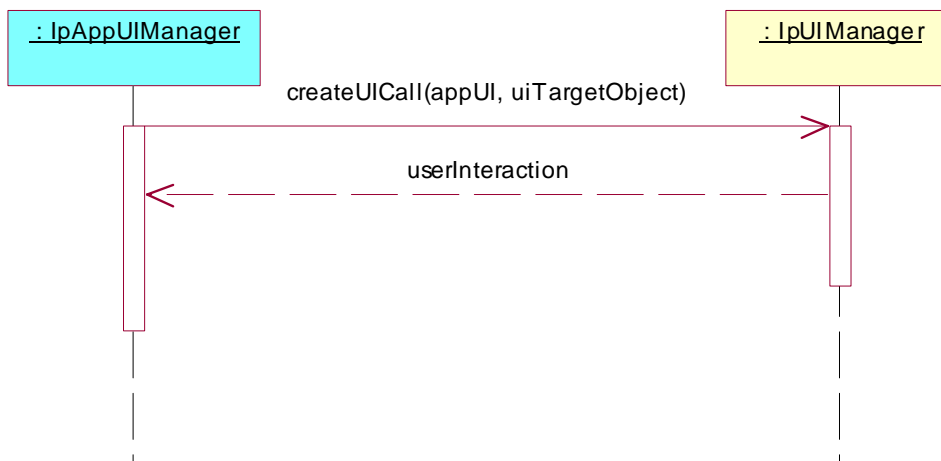Reference: ES 201 915-5 [1], clause 8.1

Selection: createUICall method supported - PICS item: [2] UIM2

Test Sequence:

1. Method call **createUICall()**
    Parameters: appUI, uiTargetObject
    Check: valid value of TpUIIdentifier is returned

Test UI_05

Summary: **IpUIManager**, createUICall method, P_INVALID_NETWORK_STATE

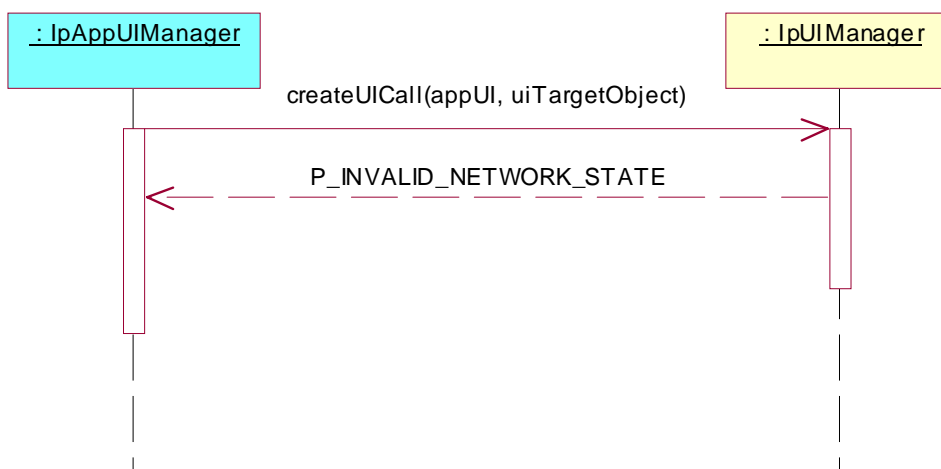Reference: ES 201 915-5 [1], clause 8.1

Selection: createUICall method supported - PICS item: [2] UIM2

Condition: the underlying protocol cannot support the method call.

Test Sequence:

1.   Method call **createUICall()**
      Parameters: appUI, userAddress
      Check: P_INVALID_NETWORK_STATE is returned

   NOTE: Any other suitable exception is accepted.

```
      ┌─────────────────┐                                      ┌─────────────────┐
      │ : IpAppUIManager│                                      │  : IpUIManager  │
      └────────┬────────┘                                      └────────┬────────┘
               │            createUICall(appUI, uiTargetObject)          │
               ├───────────────────────────────────────────────────────>│
               │                                                         │
               │             P_INVALID_NETWORK_STATE                     │
               │<─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─┤
               │                                                         │
               │                                                         │
               │                                                         │
```

Test UI_06

Summary:	**IpUIManager**, createUICall method, P_INVALID_INTERFACE_TYPE

Reference:	ES 201 915-5 [1], clause 8.1

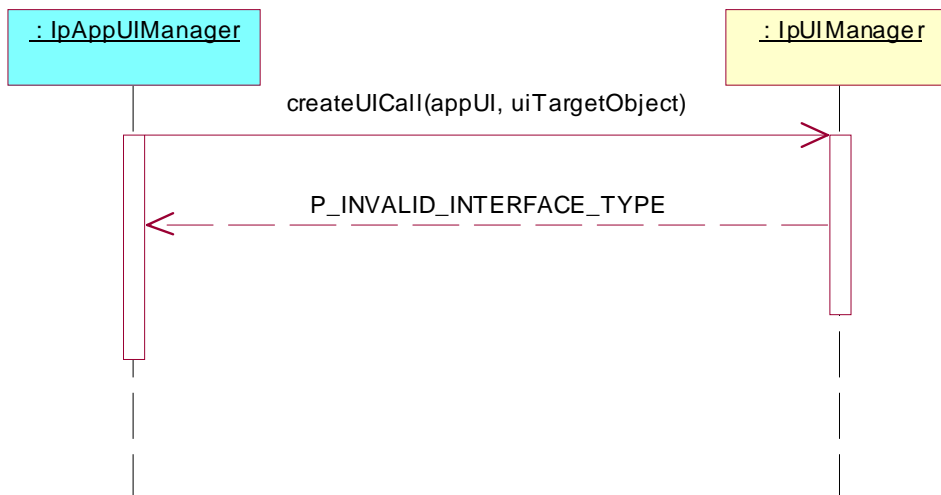Selection:	createUICall method supported - PICS item: [2] UIM2

Test Sequence:

1.	Method call **createUICall()**
	Parameters:	appUI including an invalid reference type, userAddress
	Check:		P_INVALID_INTERFACE_TYPE is returned

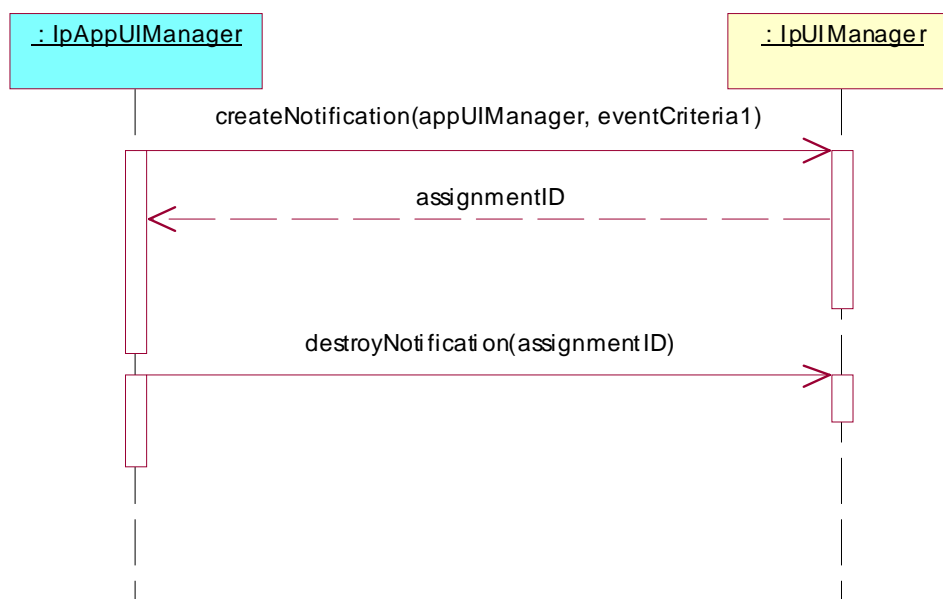NOTE:	Any other suitable exception is accepted.

Test UI_07

Summary: **IpUIManager**, all createNotification and destroyNotification methods, successful

Reference: ES 201 915-5 [1], clause 8.1

Selection: createNotification methods supported - PICS item: [2] UIM3

Test Sequence:

1. Method call **createNotification()**
   Parameters: appUIManager, eventCriteria,
   Check: valid value of TpAssignmentID is returned

2. Method call **destroyNotification()**
   Parameters: assignmentID as returned in 1..
   Check: no exception is returned.
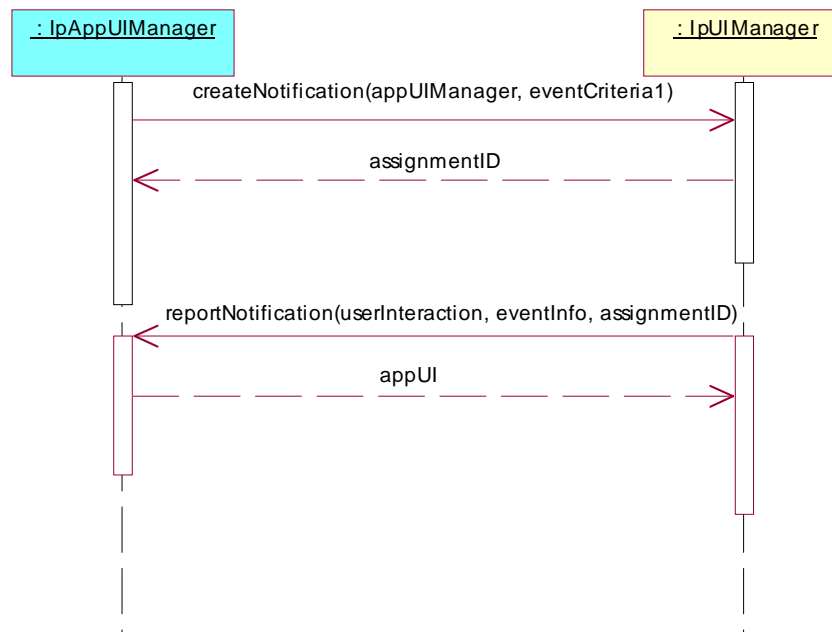
Test UI_08

Summary: **IpUIManager**, receipt of reportNotification method, successful

Reference: ES 201 915-5 [1], clauses 8.1 and 8.2

Selection: createNotification methods supported - PICS item: [2] UIM3

Test Sequence:

1.   Method call **createNotification()**
     Parameters: appUIManager, eventCriteria,
     Check: valid value of TpAssignmentID is returned

2.   Triggered action: cause IUT to call **reportNotification()** method on the tester's (Application)
     IpAppUIManager interface.
     Check: valid value of userInteraction, eventInfo and assignmentID given in 1. is transmitted,
                      Return: valid value of IpAppUIRef

Test UI_09

Summary: **IpUIManager**, createNotification method, P_INVALID_INTERFACE_TYPE

Reference: ES 201 915-5 [1], clause 8.1

Selection: createNotification methods supported - PICS item: [2] UIM3
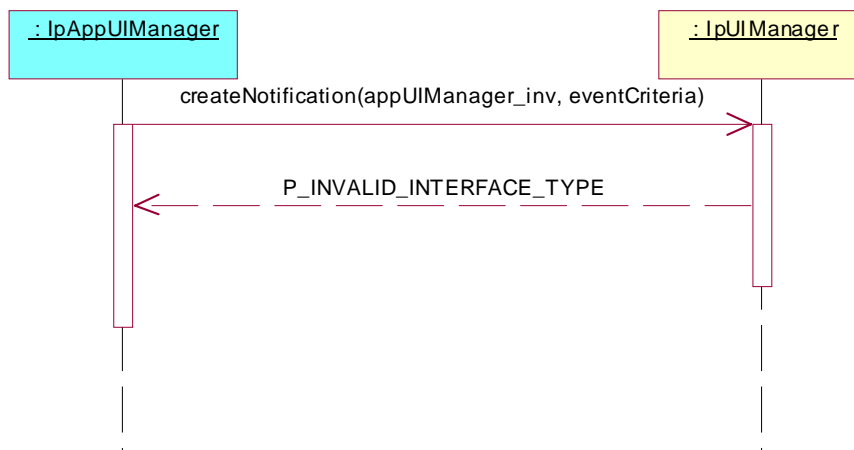
Test Sequence:

   1.   Method call **createNotification()**
       Parameters: appUIManager including an invalid reference type, eventCriteria,
       Check: P_INVALID_INTERFACE_TYPE is returned

   NOTE: Any other suitable exception is accepted.

Test UI_10

Summary: **IpUIManager**, createNotification method, P_INVALID_CRITERIA

Reference: ES 201 915-5 [1], clause 8.1

Selection: createNotification methods supported - PICS item: [2] UIM3
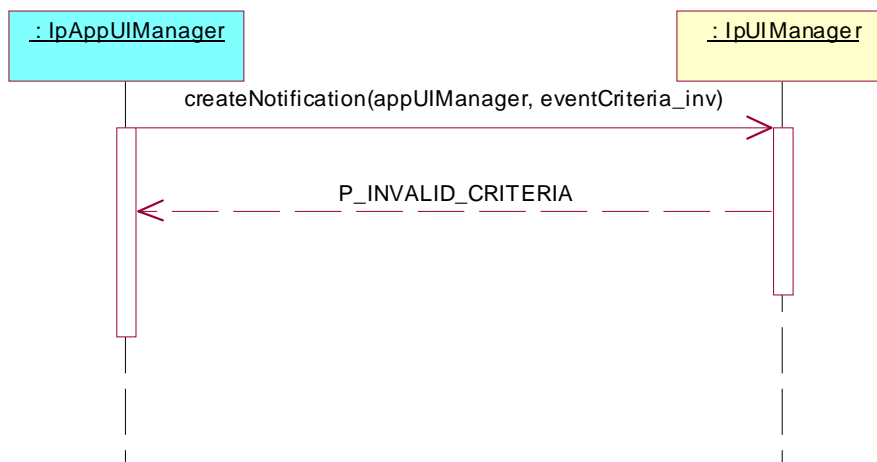
Test Sequence:

1.   Method call **createNotification()**
      Parameters: appUIManager, eventCriteria containing an invalid criteria.
      Check: P_INVALID_CRITERIA is returned.

   NOTE: Any other suitable exception is accepted.

Test UI_11

Summary: **IpUIManager**, createNotification and changeNotification methods, P_INVALID_ASSIGNMENT_ID
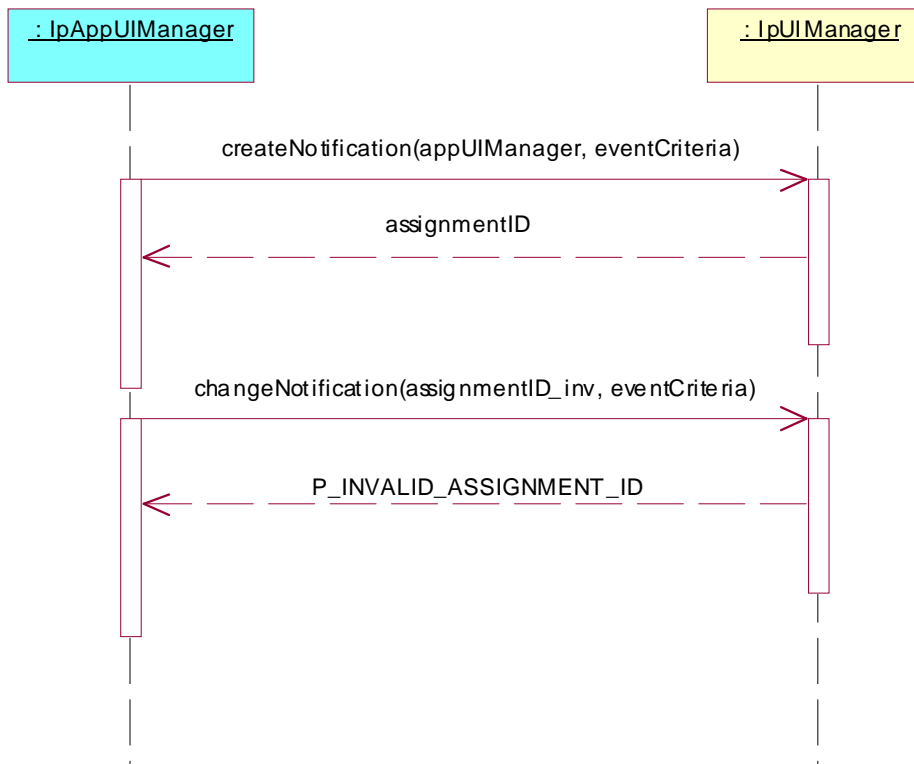
Reference: ES 201 915-5 [1], clause 8.1

Selection: createNotification and changeNotification methods supported - PICS item: [2] UIM3 AND UIM5

Test Sequence:

1.   Method call **createNotification()**
     Parameters: appUIManager, eventCriteria,
     Check: valid value of TpAssignmentID is returned

2.   Method call **changeNotification()**
     Parameters: assignmentID different from returned in 1., eventCriteria.
     Check: P_INVALID_ASSIGNMENT_ID is returned.

NOTE: Any other suitable exception is accepted.

Test UI_12

Summary: **IpUIManager**, createNotification and changeNotification method, P_INVALID_CRITERIA
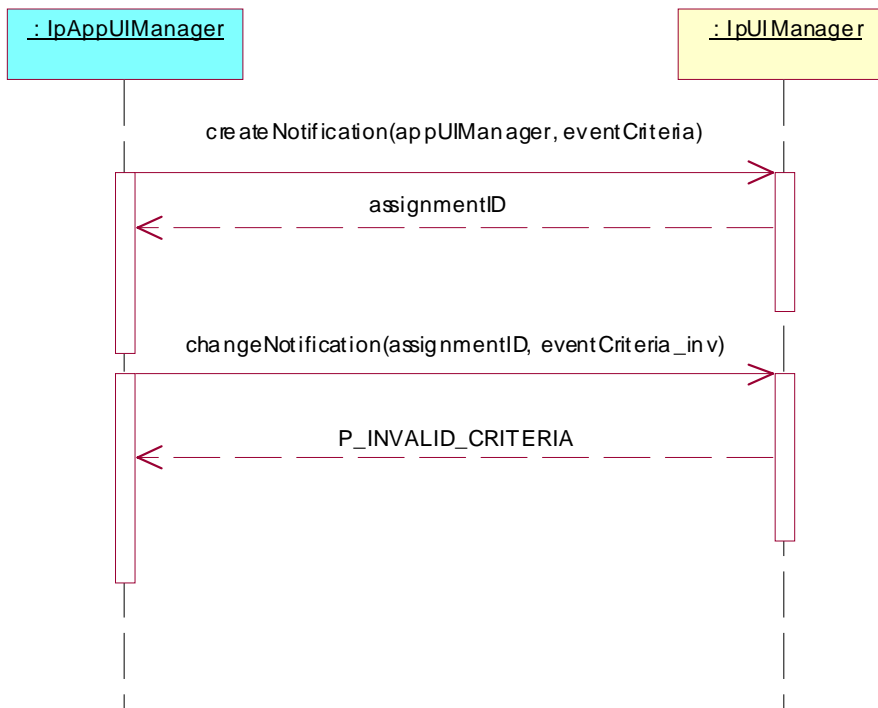
Reference: ES 201 915-5 [1], clause 8.1

Selection: createNotification and changeNotification methods supported - PICS item: [2] UIM3 AND UIM5

Test Sequence:

1.   Method call **createNotification()**
     Parameters: appUIManager, eventCriteria,
     Check: valid value of TpAssignmentID is returned

2.   Method call **changeNotification()**
     Parameters: assignmentID as returned in 1., eventCriteria containing an invalid criteria..
     Check: P_INVALID_CRITERIA is returned.

NOTE: Any other suitable exception is accepted.

Test UI_13

Summary: **IpUIManager**, destroyNotification method, P_INVALID_ASSIGNMENT_ID
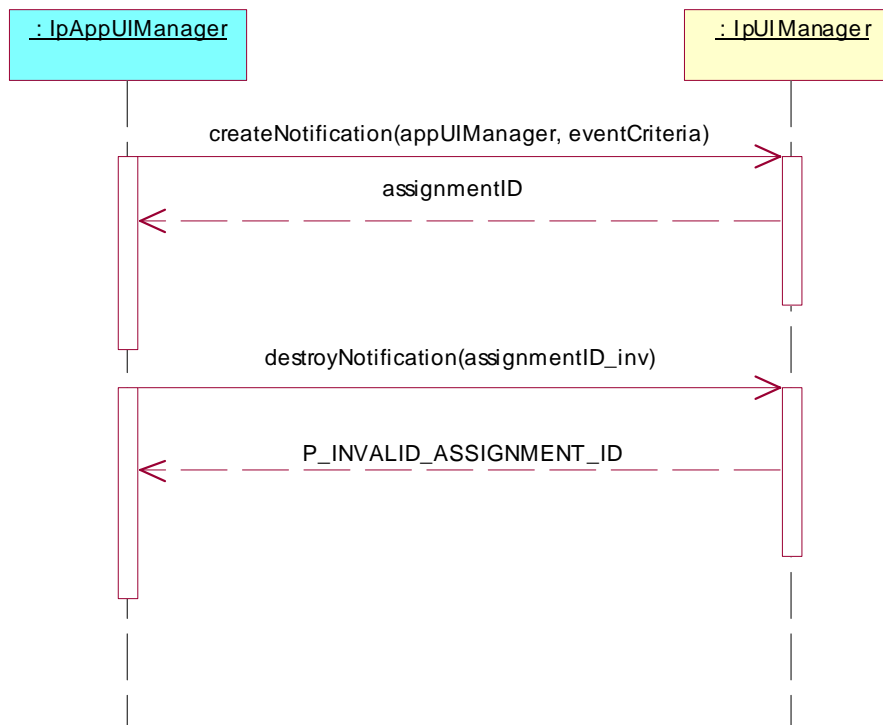
Reference: ES 201 915-5 [1], clause 8.1

Selection: createNotification method supported - PICS item: [2] UIM3

Test Sequence:

1. Method call **createNotification()**
    Parameters: appUIManager, eventCriteria,
    Check: valid value of TpAssignmentID is returned

2. Method call **destroyNotification()**
    Parameters: assignmentID different from 1..
    Check: P_INVALID_ASSIGNMENT_ID is returned.

NOTE: Any other suitable exception is accepted.

5.2.1.2   IpUI interface

Test UI_14

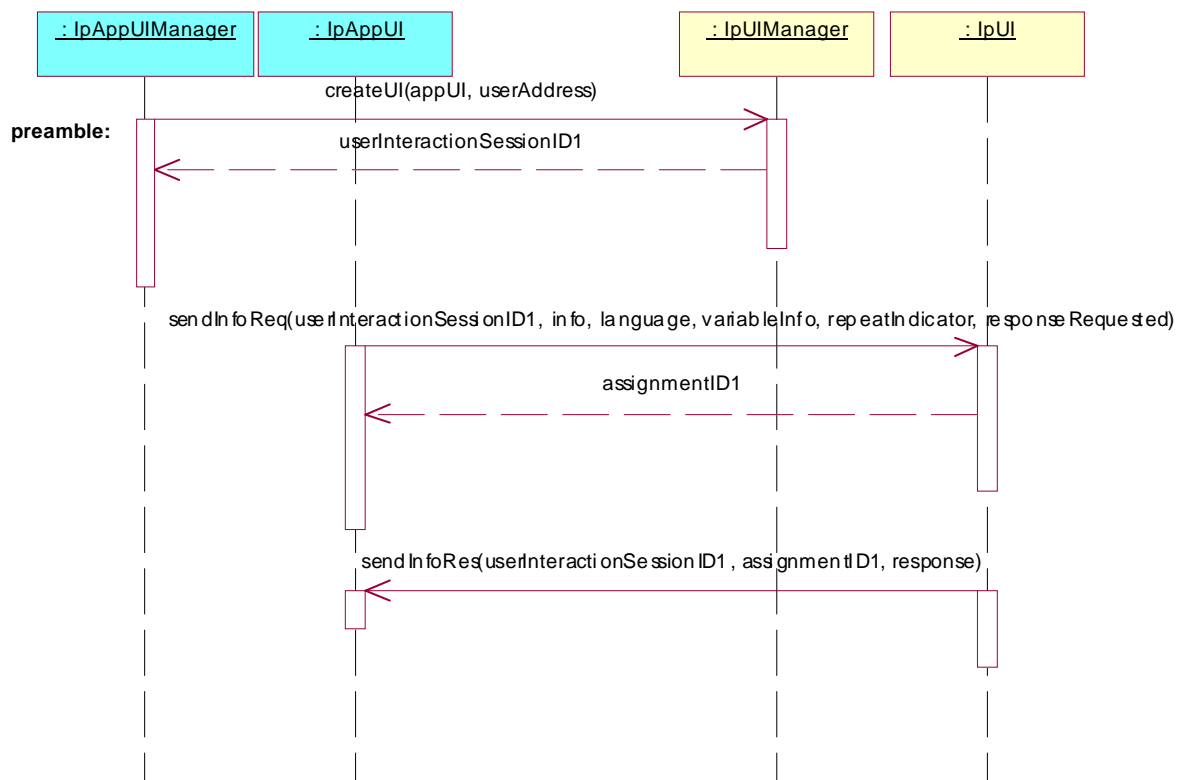Summary: **IpUI**, sendInfoReq & sendInfoRes methods, successful

Reference: ES 201 915-5 [1], clauses 8.3 and 8.4

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                 IpUIManager interface or reported in a reportNotification method, exists. The given sessionID is used
                 in the following methods to identify this User Interaction object.

Selection: sendInfoReq method supported - PICS item: [2] UI1

Test Sequence:

1.   Method call **sendInfoReq()**
     Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, repeatIndicator,
                      responseRequested.
     Check: valid value of TpAssignmentID is returned

2.   Triggered action: cause IUT to call **sendInfoRes()** method on the tester's (Application) **IpAppUI** interface.
     Check: valid value of userInteractionSessionID (given in preamble), assignmentID (given in 1.) and
                      response is transmitted,
     Return: no parameter

Test UI_15

Summary: **IpUI**, sendInfoReq methods, P_INVALID_SESSION_ID

Reference: ES 201 915-5 [1], clause 8.3

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
               IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
               the following methods to identify this User Interaction object.

Selection: sendInfoReq method supported - PICS item: [2] UI1
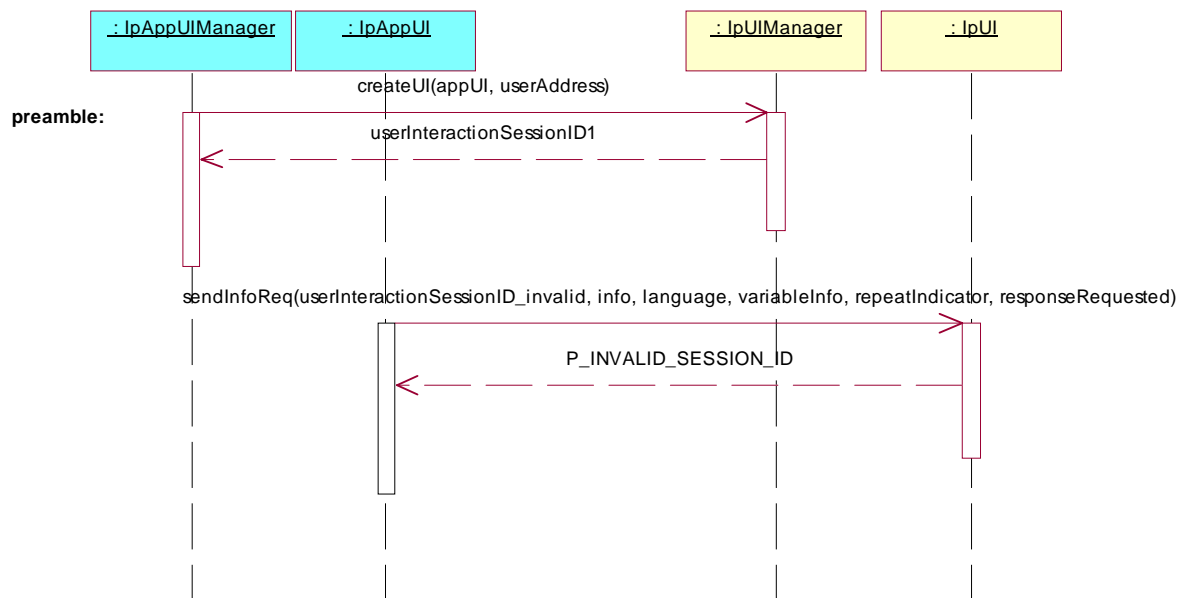
Test Sequence:

1.   Method call **sendInfoReq()**
     Parameters: invalid userInteractionSessionID, info, language, variableInfo, repeatIndicator,
                    responseRequested.
     Check: P_INVALID_SESSION_ID is returned

     NOTE: Any other suitable exception is accepted.

Test UI_16

Summary: **IpUI**, sendInfoReq methods, P_INVALID_NETWORK_STATE

Reference: ES 201 915-5 [1], clause 8.3

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                  IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                  the following methods to identify this User Interaction object.

Condition: the underlying protocol cannot support the method call.

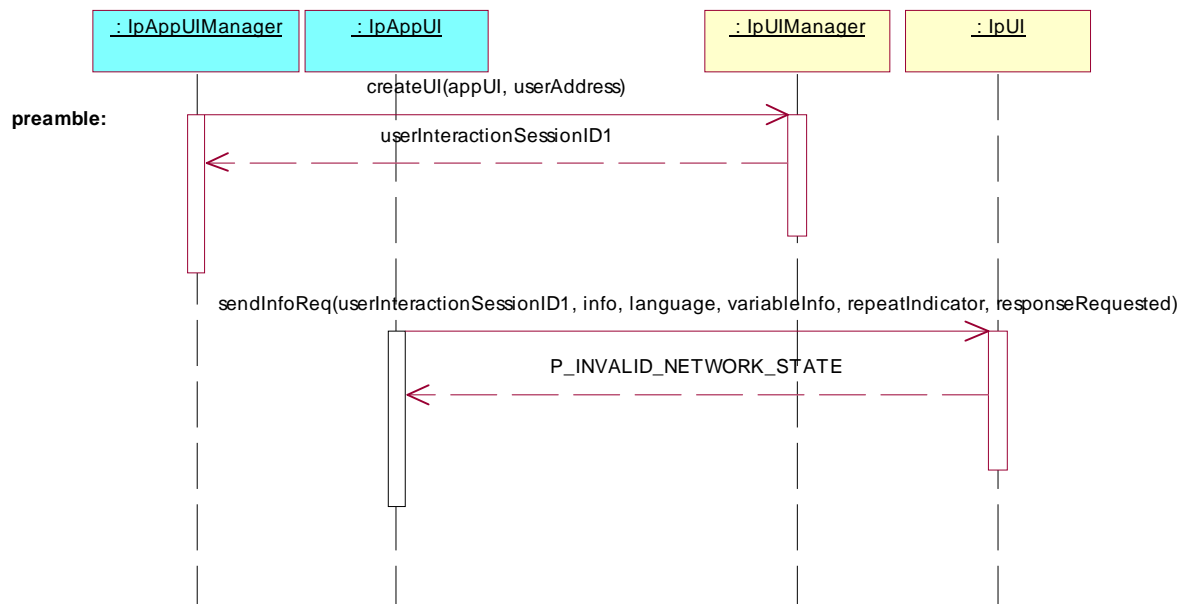Selection: sendInfoReq method supported - PICS item: [2] UI1

Test Sequence:

   1.   Method call **sendInfoReq()**
        Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, repeatIndicator,
                         responseRequested.
        Check: P_INVALID_NETWORK_STATE is returned

   NOTE: Any other suitable exception is accepted.

Test UI_17

Summary: **IpUI**, sendInfoReq methods, P_ILLEGAL_ID

Reference: ES 201 915-5 [1], clause 8.3

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in the following methods to identify this User Interaction object.

Selection: sendInfoReq method supported - PICS item: [2] UI1
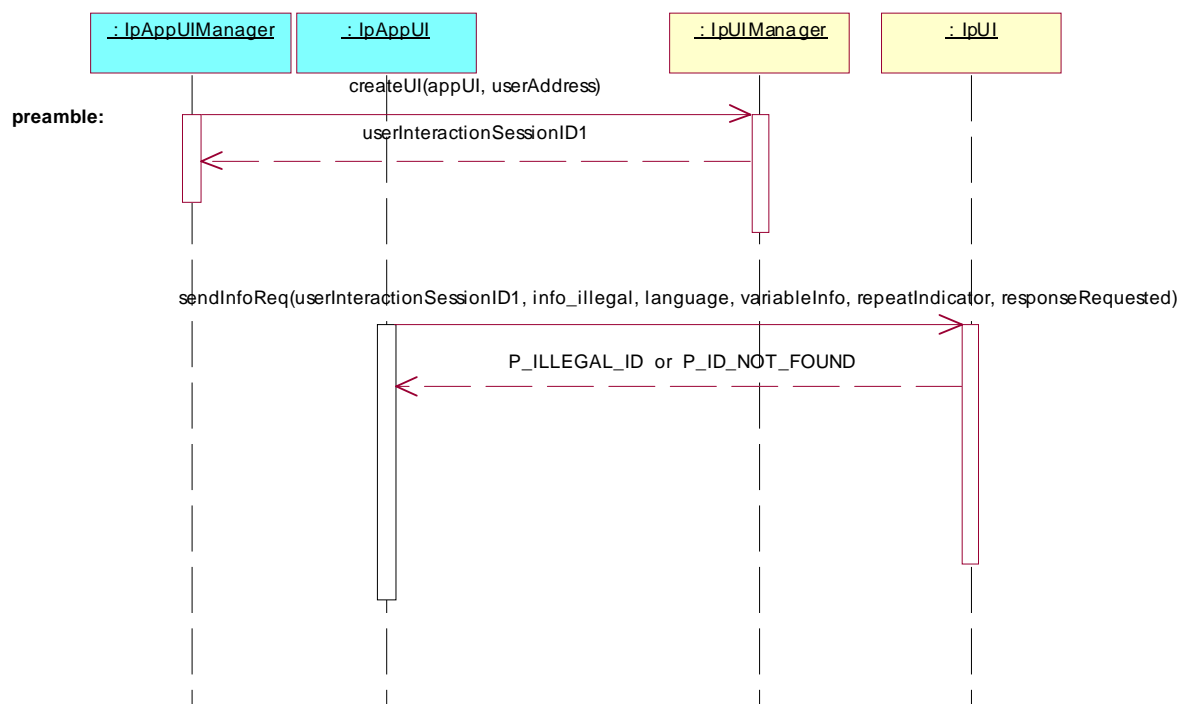
Test Sequence:

1.   Method call **sendInfoReq()**
       Parameters: userInteractionSessionID given in preamble, illegal info, language, variableInfo, repeatIndicator, responseRequested.
       Check: P_ILLEGAL_ID or P_ID_NOT_FOUND is returned

NOTE: Any other suitable exception is accepted.

Test UI_18

Summary: **IpUI**, sendInfoAndCollectReq & sendInfoAndCollectRes methods, successful
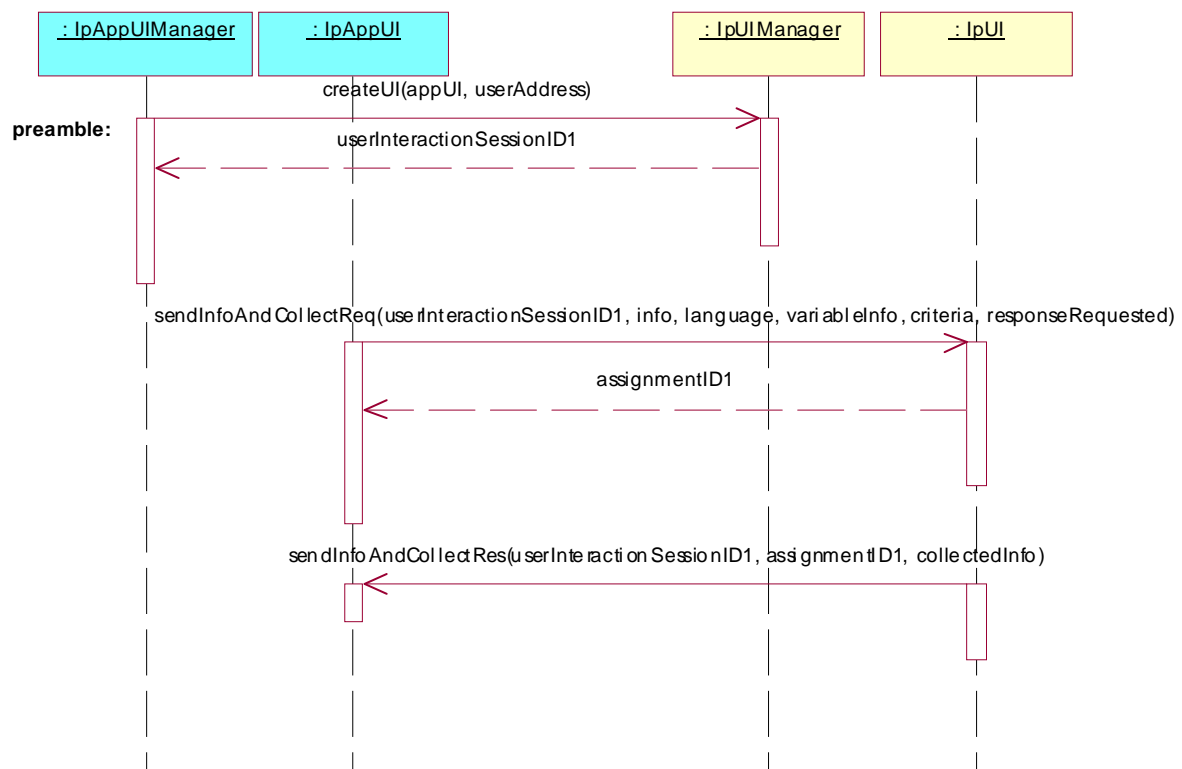
Reference: ES 201 915-5 [1], clauses 8.3 and 8.4

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

Selection: sendInfoAndCollectReq method supported - PICS item: [2] UI2

Test Sequence:

1. Method call **sendInfoAndCollectReq()**
    Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, criteria,
                    responseRequested.
    Check: valid value of TpAssignmentID is returned

2. Triggered action: cause IUT to call **sendInfoAndCollectRes()** method on the tester's (Application) **IpAppUI**
    interface.
    Check: valid value of userInteractionSessionID (given in preamble), assignmentID (given in 1.),
                    response and collectedInfo is transmitted,
    Return: no parameter

Test UI_19

Summary: **IpUI**, sendInfoAndCollectReq methods, P_INVALID_SESSION_ID

Reference: ES 201 915-5 [1], clause 8.3

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

Selection: sendInfoAndCollectReq method supported - PICS item: [2] UI2
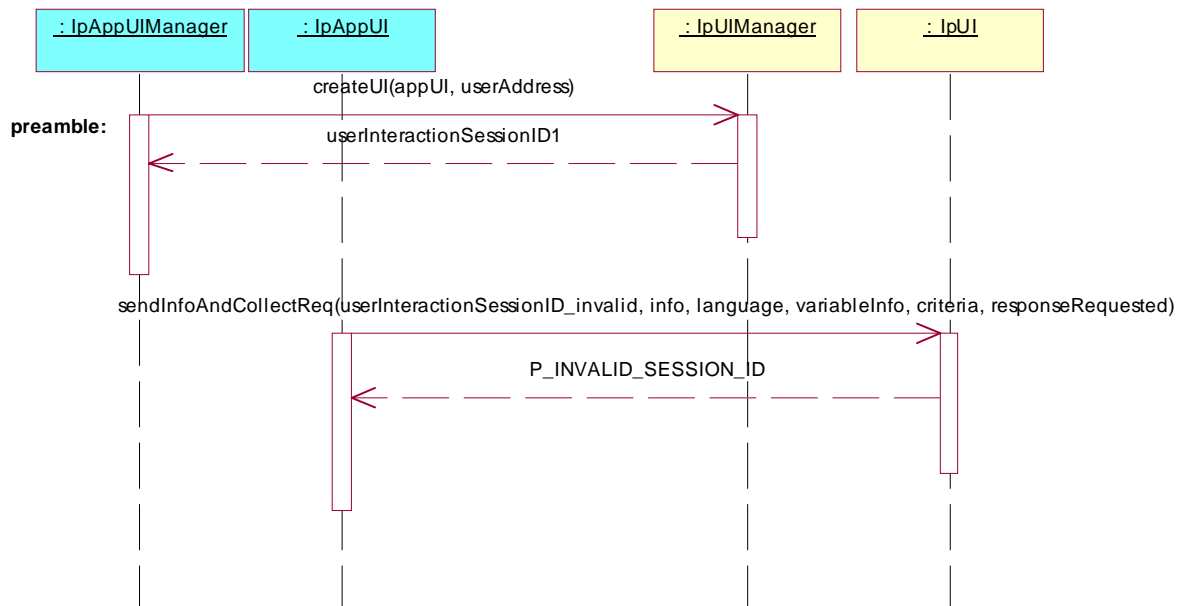
Test Sequence:

1.   Method call **sendInfoAndCollectReq()**
     Parameters: invalid userInteractionSessionID, info, language, variableInfo, criteria, responseRequested.
     Check: P_INVALID_SESSION_ID is returned

NOTE: Any other suitable exception is accepted.

Test UI_20

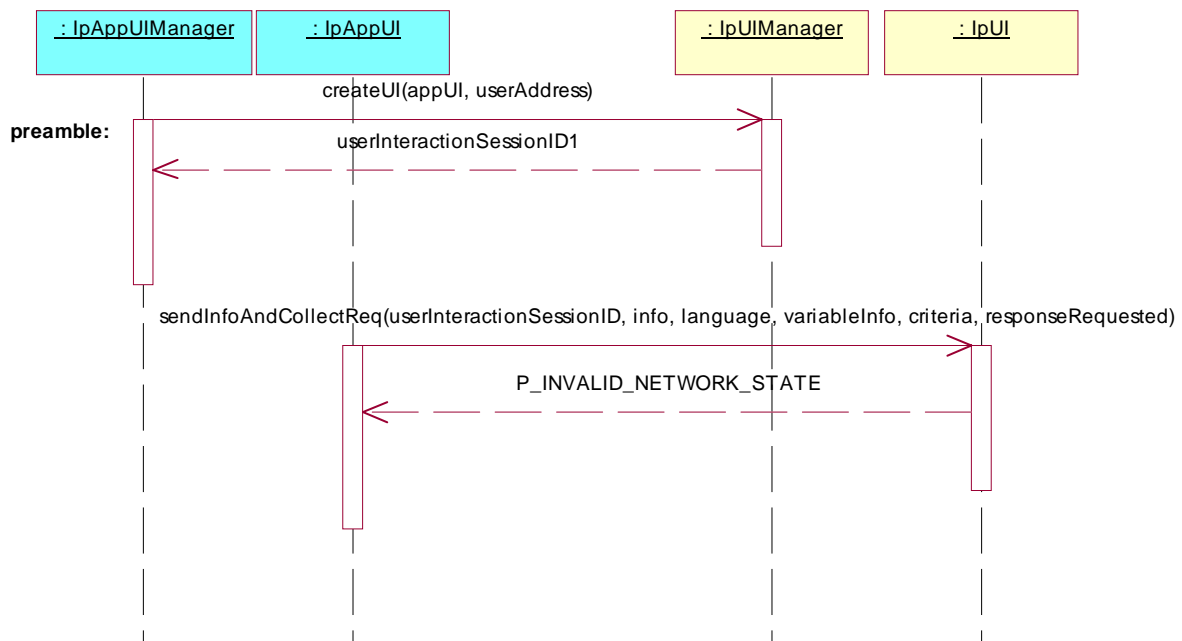|	
Summary:	**IpUI**, sendInfoAndCollectReq methods, P_INVALID_NETWORK_STATE
Reference:	ES 201 915-5 [1], clause 8.3
Preamble:	A User Interaction object, created either with the createUI or the createUICall method on the IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in the following methods to identify this User Interaction object.
Condition:	the underlying protocol cannot support the method call.
Selection:	sendInfoAndCollectReq method supported - PICS item: [2] UI2

Test Sequence:

1. Method call **sendInfoAndCollectReq()**
   Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, criteria, responseRequested.
   Check: P_INVALID_NETWORK_STATE is returned

NOTE: Any other suitable exception is accepted.

Test UI_21

Summary: **IpUI**, sendInfoAndCollectReq methods, P_ILLEGAL_ID

Reference: ES 201 915-5 [1], clause 8.3

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
               IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
               the following methods to identify this User Interaction object.

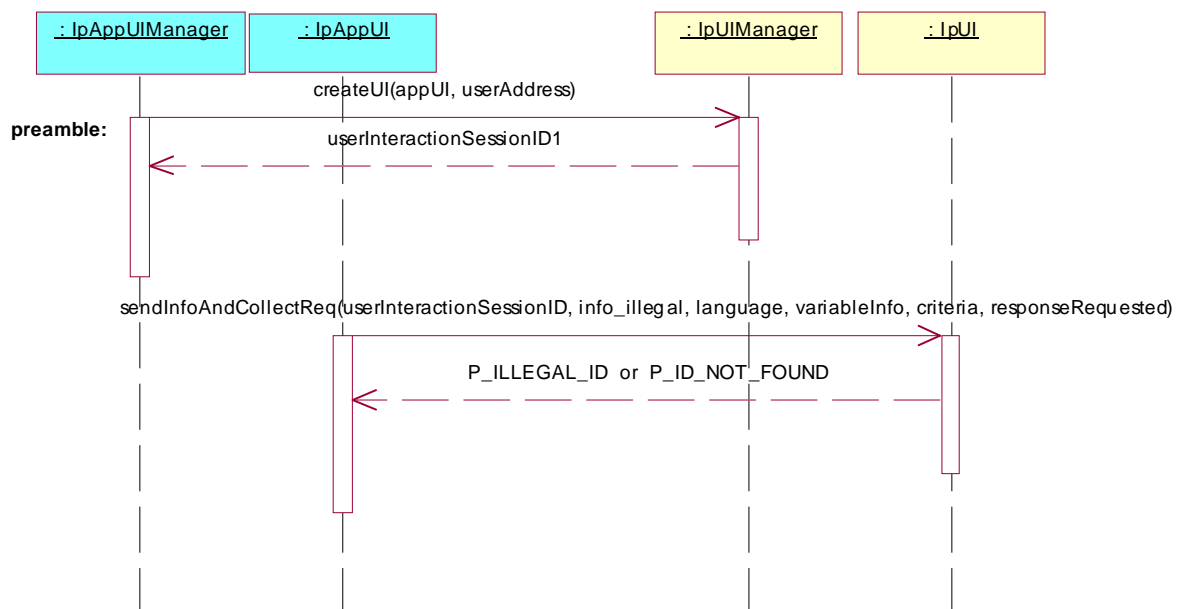Selection: sendInfoAndCollectReq method supported - PICS item: [2] UI2

Test Sequence:

1.   Method call **sendInfoAndCollectReq()**
     Parameters: userInteractionSessionID given in preamble, illegal info, language, variableInfo, criteria,
                     responseRequested
     Check: P_ILLEGAL_ID or P_ID_NOT_FOUND is returned

NOTE: Any other suitable exception is accepted.

Test UI_22

Summary: **IpUI**, sendInfoAndCollectReq methods, P_INVALID_CRITERIA

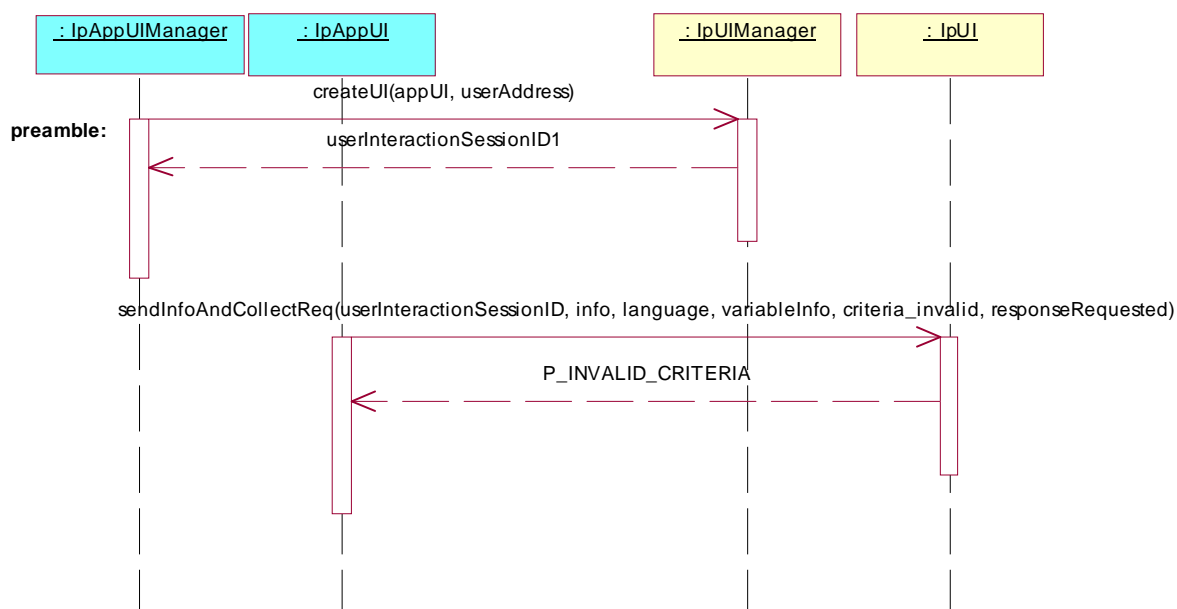Reference: ES 201 915-5 [1], clause 8.3

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in the following methods to identify this User Interaction object.

Selection: sendInfoAndCollectReq method supported - PICS item: [2] UI2

Test Sequence:

1.   Method call **sendInfoAndCollectReq()**
     Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, invalid criteria, responseRequested.
     Check: P_INVALID_CRITERIA is returned

NOTE: Any other suitable exception is accepted.

Test UI_23

Summary: **IpUI**, sendInfoAndCollectReq methods, P_ILLEGAL_RANGE

Reference: ES 201 915-5 [1], clause 8.3

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

Selection: sendInfoAndCollectReq method supported - PICS item: [2] UI2
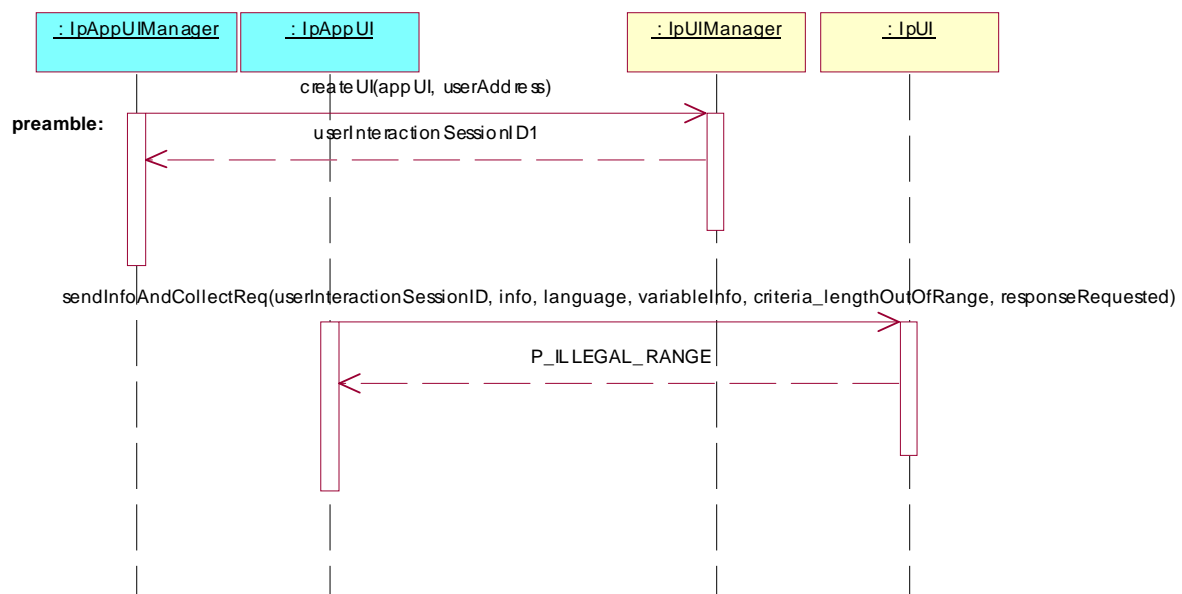
Test Sequence:

   1.   Method call **sendInfoAndCollectReq()**
        Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, criteria with
                        collection length out of range, responseRequested.
        Check: P_ILLEGAL_RANGE is returned

   NOTE: Any other suitable exception is accepted.

Test UI_24

Summary: **IpUI**, sendInfoAndCollectReq methods, P_INVALID_COLLECTION_CRITERIA

Reference: ES 201 915-5 [1], clause 8.3
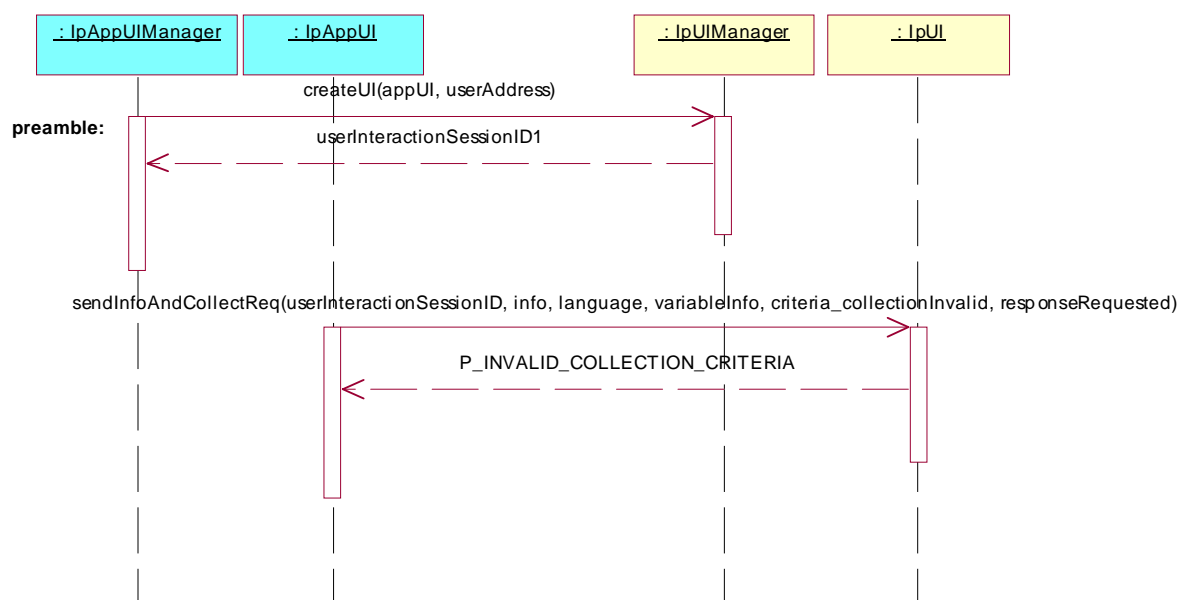
Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                  IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                  the following methods to identify this User Interaction object.

Selection: sendInfoAndCollectReq method supported - PICS item: [2] UI2

Test Sequence:

1.   Method call **sendInfoAndCollectReq()**
      Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, invalid collection
                      criteria, responseRequested.
      Check: P_INVALID_COLLECTION_CRITERIA is returned

      NOTE: Any other suitable exception is accepted.
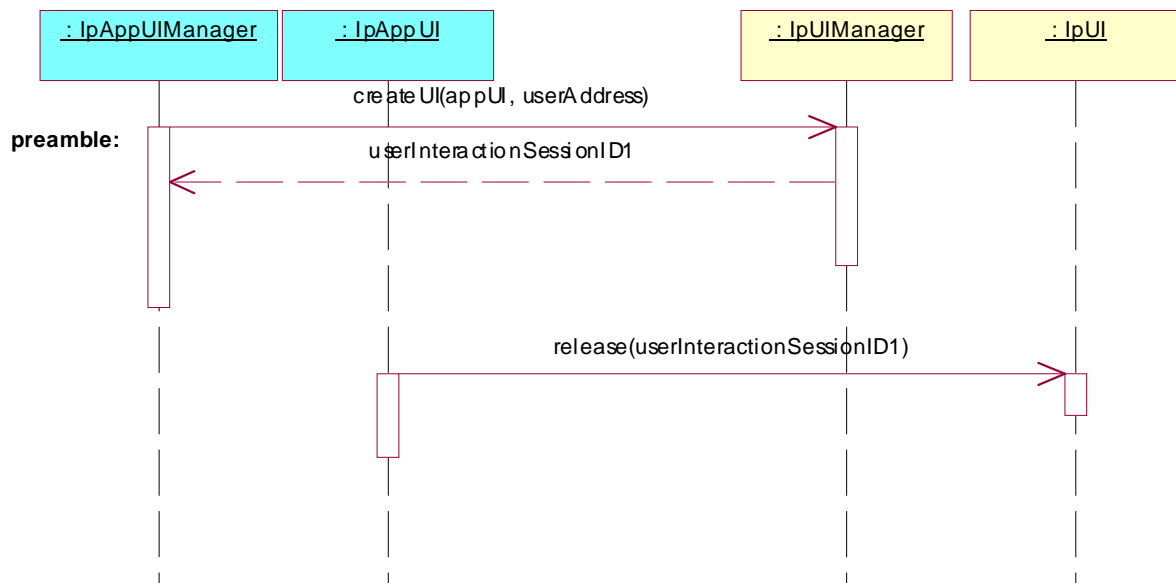
Test UI_25

Summary: **IpUI**, release method, successful

Reference: ES 201 915-5 [1], clause 8.3

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface, exists. The given sessionID is used in the following methods to identify this
                User Interaction object.

Test Sequence:

1.   Method call **release()**
     Parameters: userInteractionSessionID given in preamble.
     Check: no exception is returned.

Test UI_26

Summary: **IpUI**, release method, P_INVALID_SESSION_ID
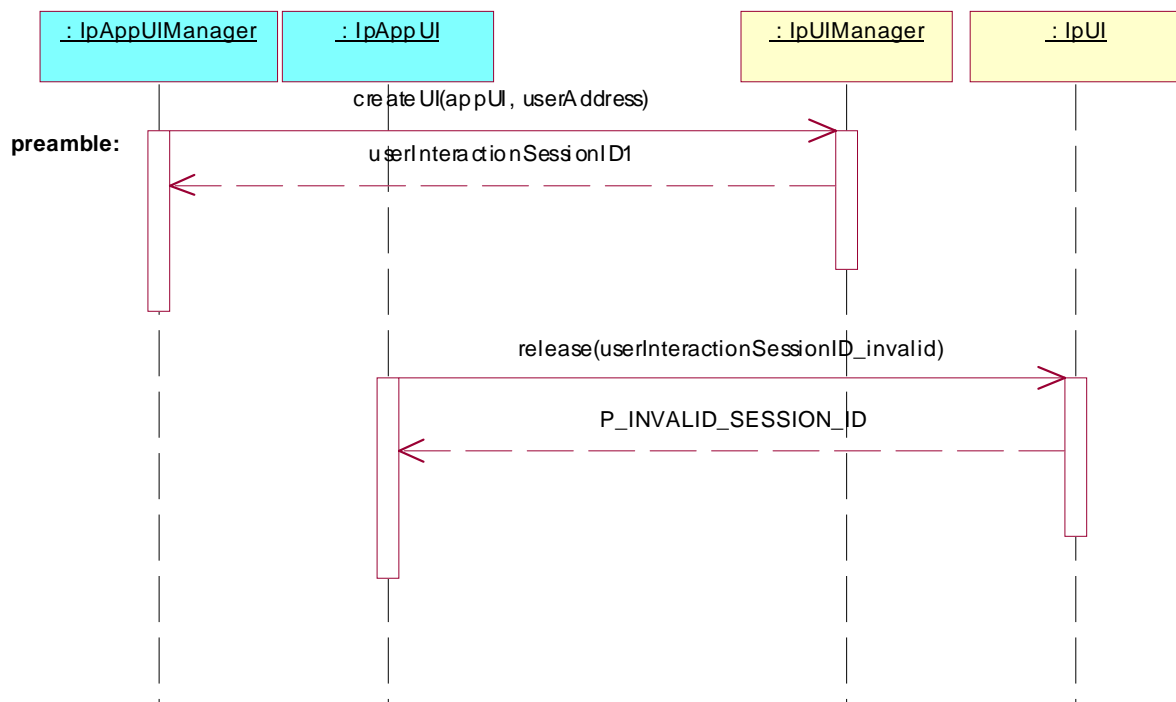
Reference: ES 201 915-5 [1], clause 8.3

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

Test Sequence:

1.   Method call **release()**
     Parameters: invalid userInteractionSessionID,
     Check: P_INVALID_SESSION_ID is returned.

NOTE: Any other suitable exception is accepted.

5.2.1.3   IpUICall interface

Test UI_27

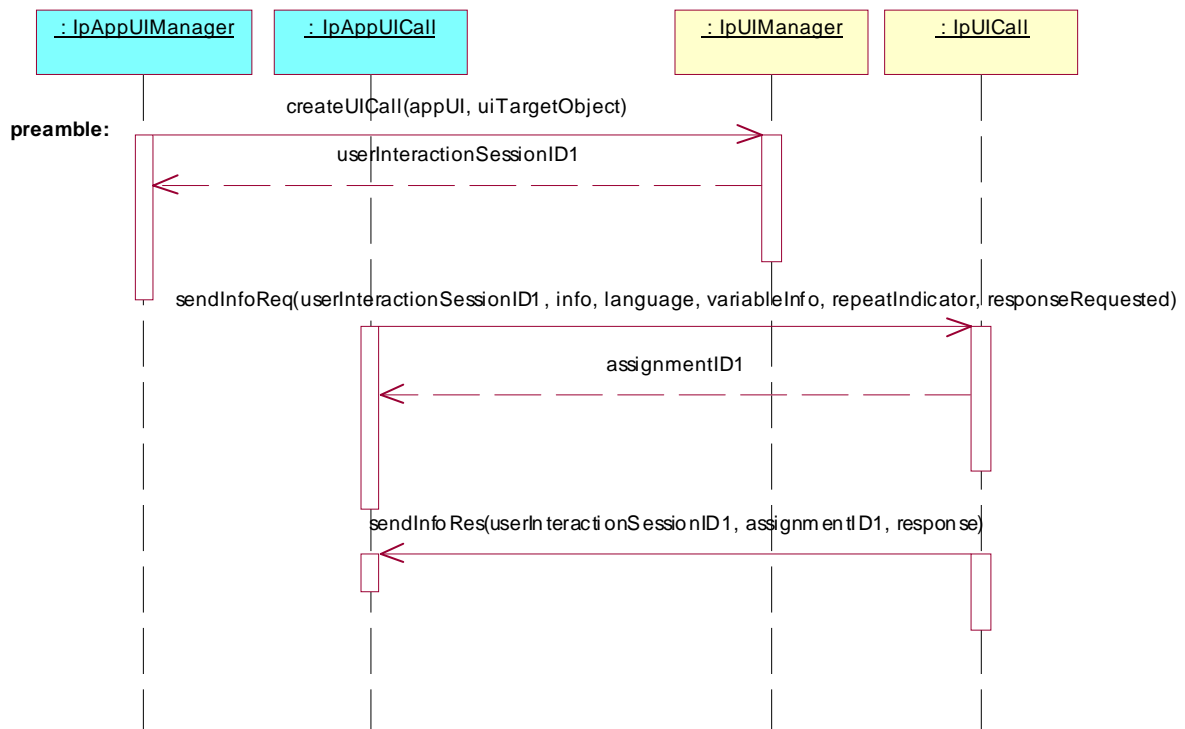Summary: **IpUICall**, sendInfoReq & sendInfoRes methods, successful

Reference: ES 201 915-5 [1], clauses 8.5 and 8.6

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method, exists. The given sessionID is used
                in the following methods to identify this User Interaction object.

Selection: sendInfoReq method supported - PICS item: [2] UIC1

Test Sequence:

1. Method call **sendInfoReq()**
    Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, repeatIndicator,
                     responseRequested.
    Check: valid value of TpAssignmentID is returned

2. Triggered action: cause IUT to call **sendInfoRes()** method on the tester's (Application) **IpAppUICall**
    interface.
    Check: valid value of userInteractionSessionID (given in preamble), assignmentID (given in 1.) and
                     response is transmitted,
    Return: no parameter

Test UI_28

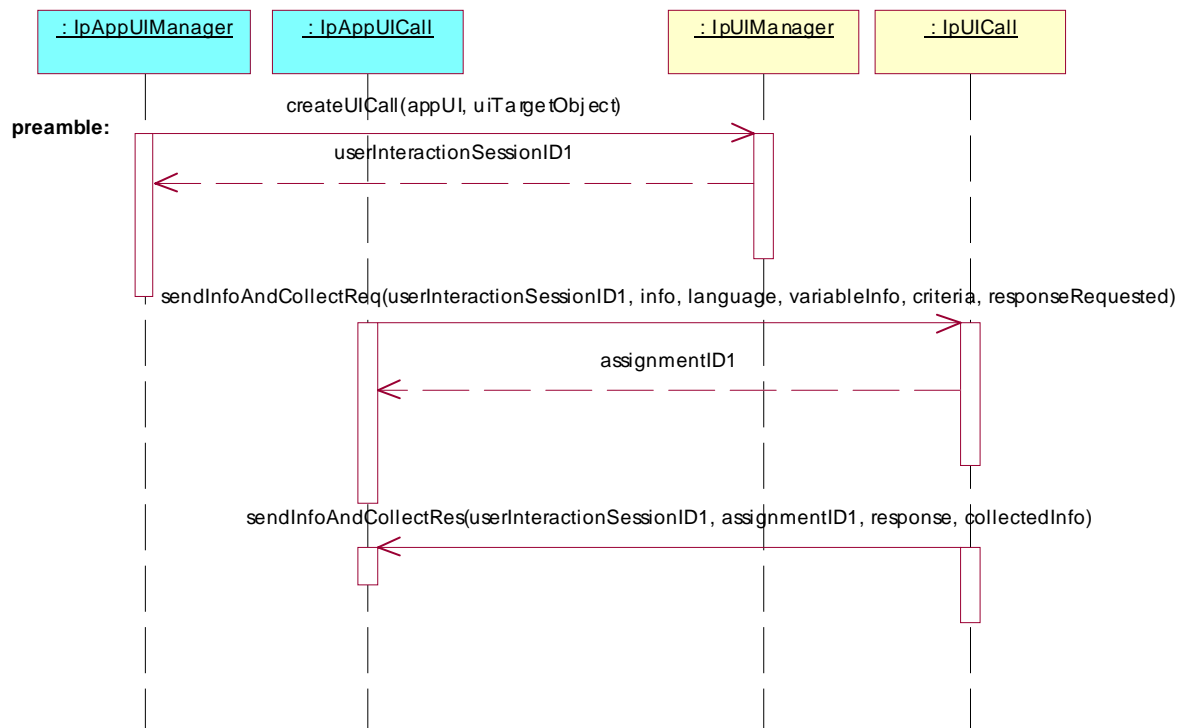|	
Summary:	**IpUICall**, sendInfoAndCollectReq & sendInfoAndCollectRes methods, successful
Reference:	ES 201 915-5 [1], clauses 8.5 and 8.6
Preamble:	A User Interaction object, created either with the createUI or the createUICall method on the IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in the following methods to identify this User Interaction object.
Selection:	sendInfoAndCollectReq method supported - PICS item: [2] UIC2

Test Sequence:

1. Method call **sendInfoAndCollectReq()**
    Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, criteria, responseRequested.
    Check: valid value of TpAssignmentID is returned

2. Triggered action: cause IUT to call **sendInfoAndCollectRes()** method on the tester's (Application) **IpAppUICall** interface.
    Check: valid value of userInteractionSessionID (given in preamble), assignmentID (given in 1.), response and collectedInfo is transmitted,
    Return: no parameter

Test UI_29

Summary: **IpUICall**, all methods, successful
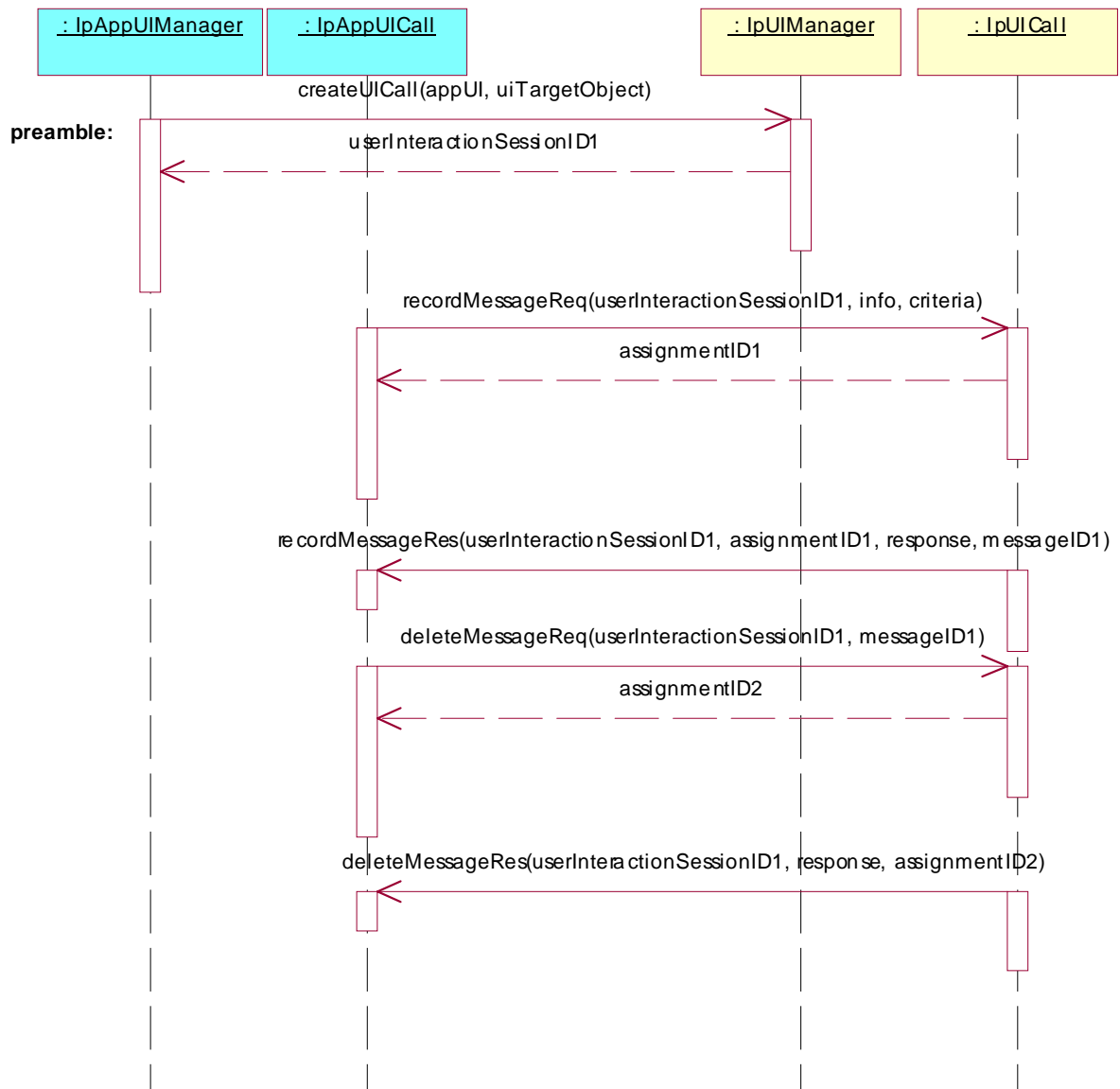
Reference: ES 201 915-5 [1], clauses 8.5 and 8.6

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

Selection: recordMessageReq and deleteMessageReq methods supported - PICS item: [2] UIC4 AND UIC5

Test Sequence:

  1. Method call **recordMessageReq()**
      Parameters: userInteractionSessionID given in preamble, info, criteria.
      Check: valid value of TpAssignmentID is returned

  2. Triggered action: cause IUT to call **recordMessageRes()** method on the tester's (Application) **IpAppUICall**
      interface.
      Check: valid value of userInteractionSessionID (given in preamble), assignmentID (given in 1.),
                    response and messageID is transmitted,
      Return: no parameter

  3. Method call **deleteMessageReq()**
      Parameters: userInteractionSessionID given in preamble, messageID returned in 2.
      Check: valid value of TpAssignmentID is returned

  4. Triggered action: cause IUT to call **deleteMessageRes()** method on the tester's (Application) **IpAppUICall**
      interface.
      Check: valid value of userInteractionSessionID (given in preamble), response and assignmentID
                    (returned in 3.) is transmitted,
      Return: no parameter

Test UI_30

Summary: **IpUICall**, abortActionReq and abortActionRes methods, successful
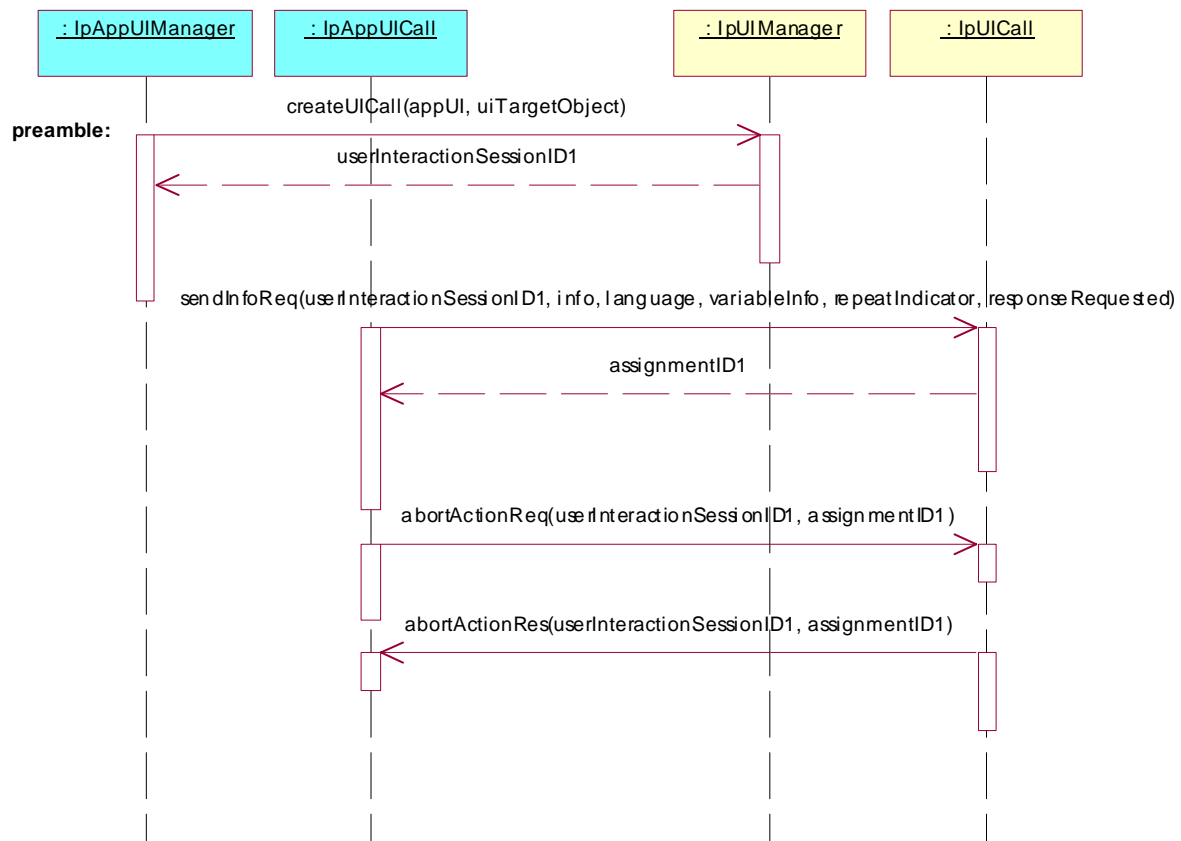
Reference: ES 201 915-5 [1], clauses 8.5 and 8.6

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

Selection: sendInfoReq and abortActionReq methods supported - PICS item: [2] UIC1 AND UIC6

Test Sequence:

1. Method call **sendInfoReq()**
    Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, repeatIndicator,
                    responseRequested.
    Check: valid value of TpAssignmentID is returned

2. Method call **abortActionReq()**
    Parameters: userInteractionSessionID given in preamble, assignmentID returned in 1.
    Check: no exception is returned.

3. Triggered action: cause IUT to call **abortActionRes()** method on the tester's (Application) **IpAppUICall**
    interface.
    Check: valid value of userInteractionSessionID (given in preamble) and assignmentID (returned in 1.)
                    is transmitted,
    Return: no parameter

Test UI_31

Summary: **IpUICall**, abortActionReq and abortActionRes methods, successful
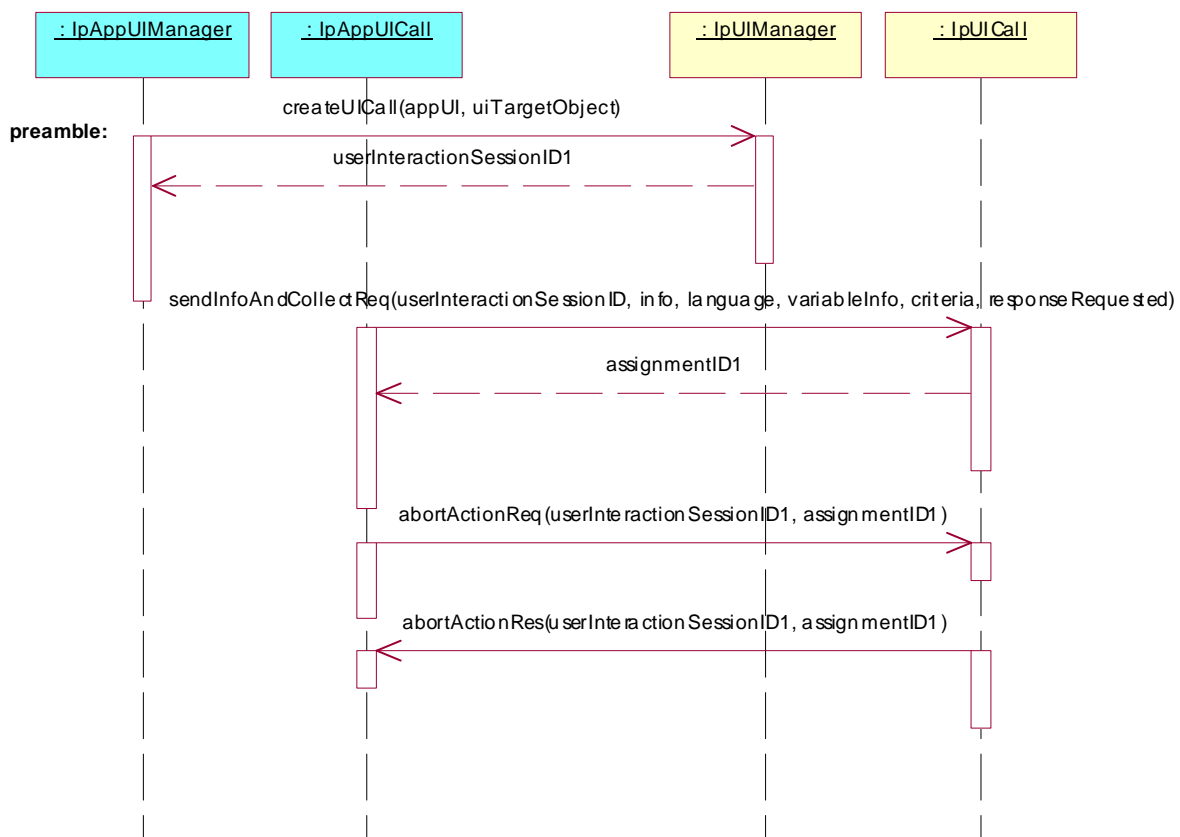
Reference: ES 201 915-5 [1], clauses 8.5 and 8.6

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in the following methods to identify this User Interaction object.

Selection: sendInfoAndCollectReq and abortActionReq methods supported - PICS item: [2] UIC2 AND UIC6

Test Sequence:

1. Method call **sendInfoAndCollectReq()**
    Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, criteria, responseRequested.
    Check: valid value of TpAssignmentID is returned

2. Method call **abortActionReq()**
    Parameters: userInteractionSessionID given in preamble, assignmentID returned in 1.
    Check: no exception is returned.

3. Triggered action: cause IUT to call **abortActionRes()** method on the tester's (Application) **IpAppUICall** interface.
    Check: valid value of userInteractionSessionID (given in preamble) and assignmentID (returned in 1.) is transmitted,
    Return: no parameter
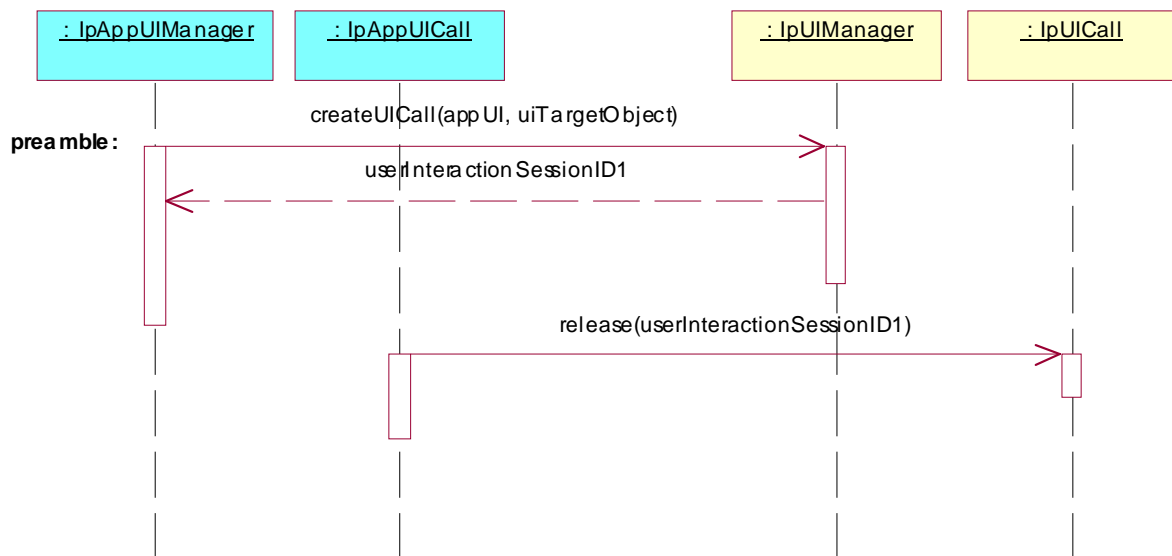
Test UI_32

Summary: **IpUICall**, release method, successful

Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createCallUIcreateUICall method on
                  the IpUIManager interface, exists. The given sessionID is used in the following methods to identify this
                  User Interaction object.

Test Sequence:

1.   Method call **release()**
      Parameters: userInteractionSessionID given in preamble.
      Check: no exception is returned.

Test UI_33

Summary: **IpUICall**, sendInfoReq methods, P_INVALID_SESSION_ID

Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

Selection: sendInfoReq method supported - PICS item: [2] UIC1
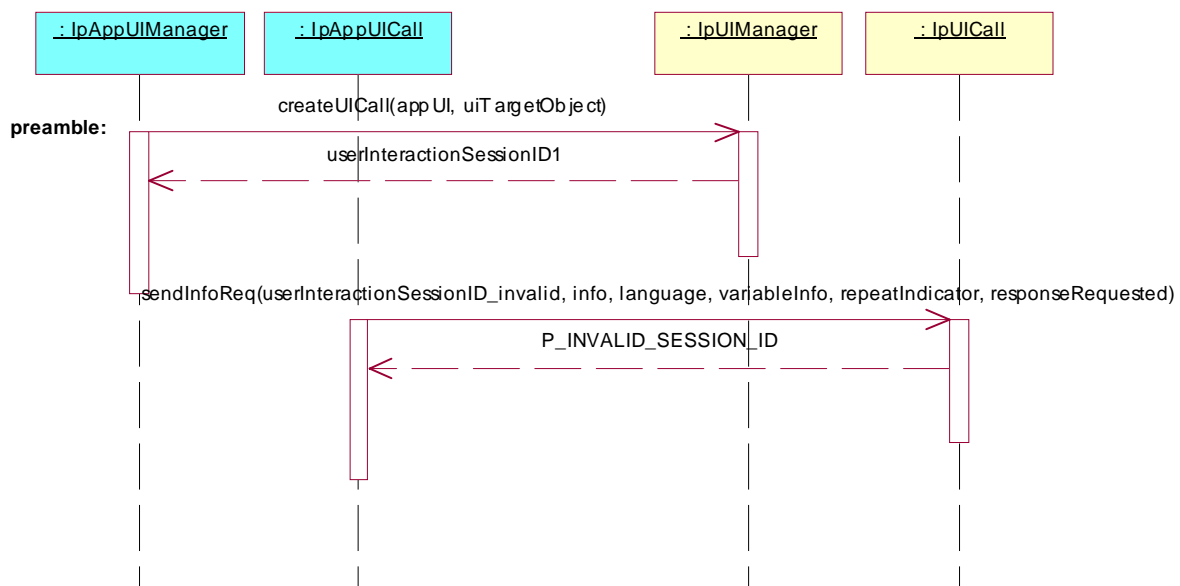
Test Sequence:

  1.   Method call **sendInfoReq()**
       Parameters: invalid userInteractionSessionID, info, language, variableInfo, repeatIndicator,
                       responseRequested.
       Check: P_INVALID_SESSION_ID is returned

  NOTE: Any other suitable exception is accepted.

Test UI_34

Summary: **IpUICall**, sendInfoReq methods, P_INVALID_NETWORK_STATE

Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in the following methods to identify this User Interaction object.

Condition: the underlying protocol cannot support the method call.

Selection: sendInfoReq method supported - PICS item: [2] UIC1
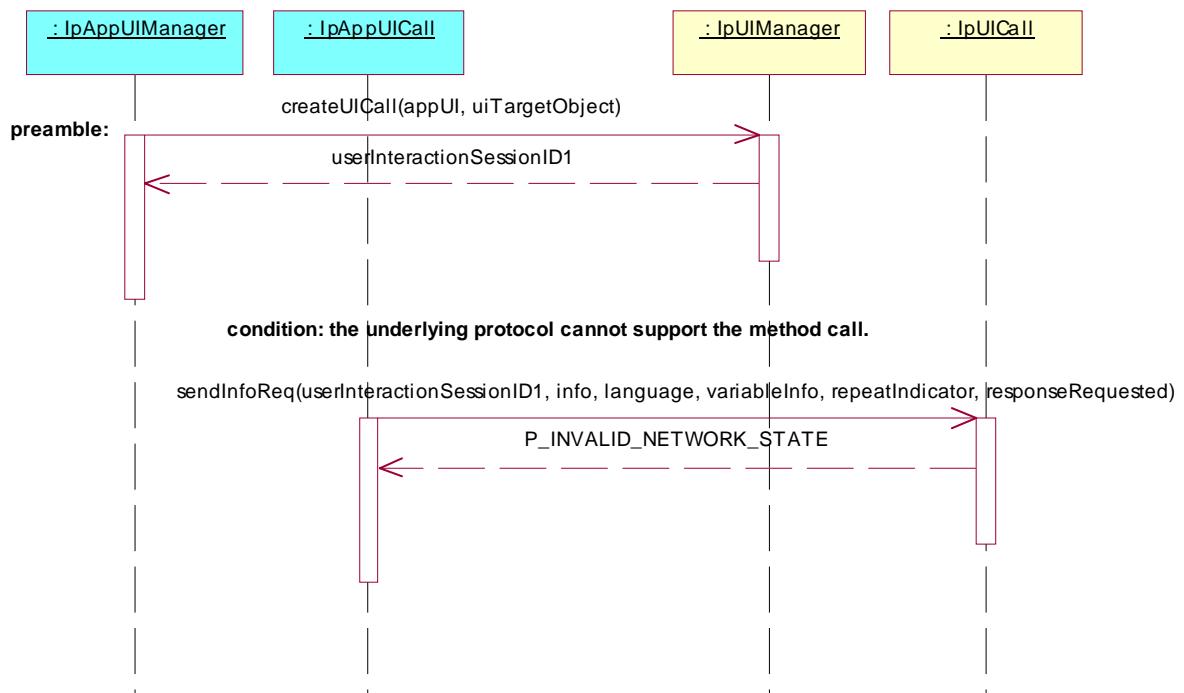
Test Sequence:

1. Method call **sendInfoReq()**
    Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, repeatIndicator, responseRequested.
    Check: P_INVALID_NETWORK_STATE is returned

NOTE: Any other suitable exception is accepted.

Test UI_35

Summary: **IpUICall**, sendInfoReq methods, P_ILLEGAL_ID

Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                  IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                  the following methods to identify this User Interaction object.

Selection: sendInfoReq method supported - PICS item: [2] UIC1
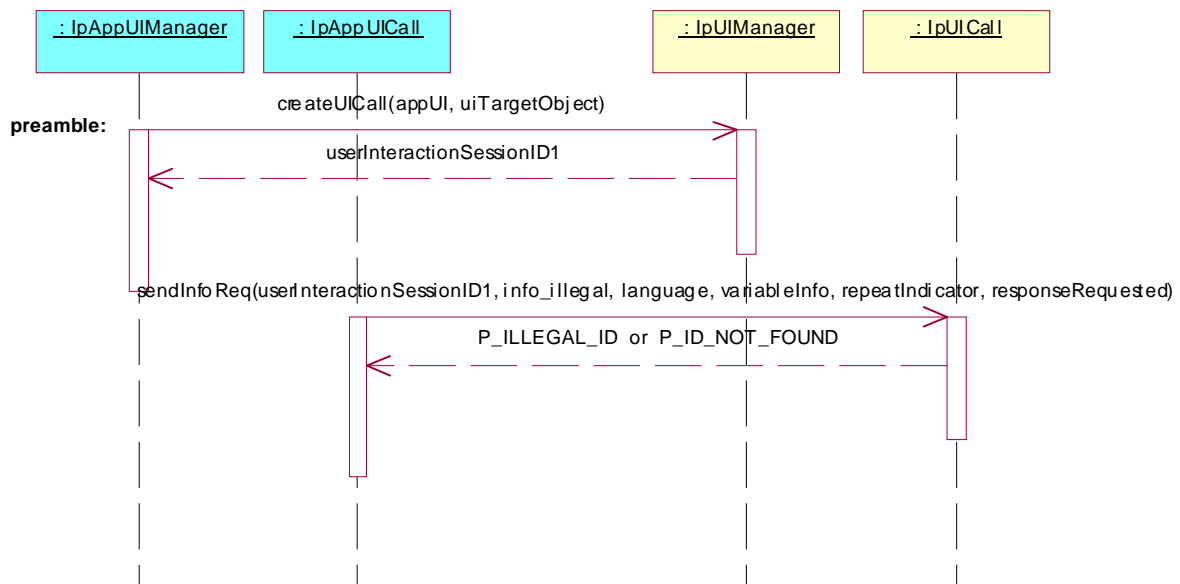
Test Sequence:

1.   Method call **sendInfoReq()**
     Parameters: userInteractionSessionID given in preamble, illegal info, language, variableInfo,
                      repeatIndicator, responseRequested.
     Check: P_ILLEGAL_ID or P_ID_NOT_FOUND is returned

NOTE: Any other suitable exception is accepted.

Test UI_36

Summary: **IpUICall**, sendInfoAndCollectReq methods, P_INVALID_SESSION_ID

Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
               IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
               the following methods to identify this User Interaction object.

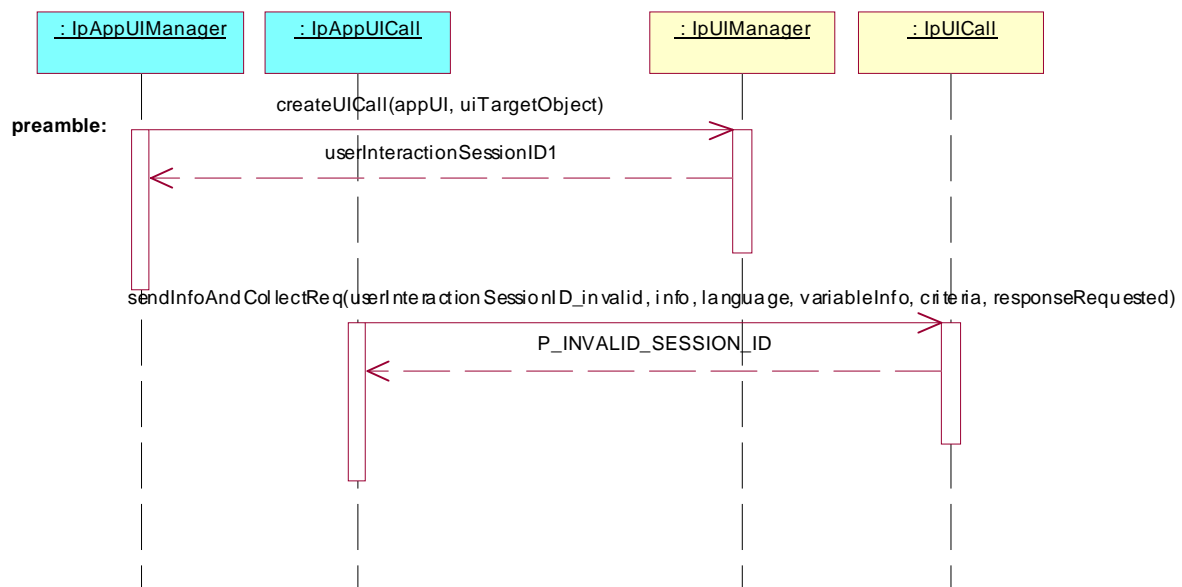Selection: sendInfoAndCollectReq method supported - PICS item: [2] UIC2

Test Sequence:

   1.   Method call **sendInfoAndCollectReq()**
        Parameters: invalid userInteractionSessionID, info, language, variableInfo, criteria, responseRequested.
        Check: P_INVALID_SESSION_ID is returned

   NOTE: Any other suitable exception is accepted.

Test UI_37

Summary: **IpUICall**, sendInfoAndCollectReq methods, P_INVALID_NETWORK_STATE

Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                  IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                  the following methods to identify this User Interaction object.
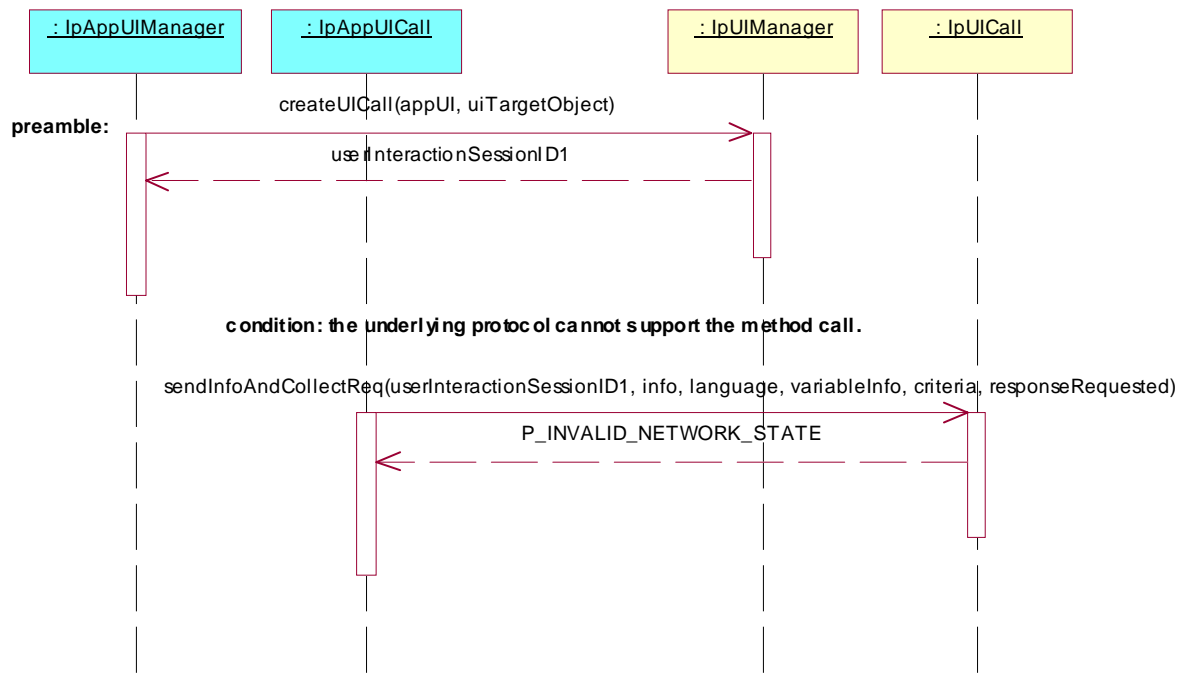
Condition: the underlying protocol cannot support the method call.

Selection: sendInfoAndCollectReq method supported - PICS item: [2] UIC2

Test Sequence:

   1.   Method call **sendInfoAndCollectReq()**
      Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, criteria,
                  responseRequested.
      Check: P_INVALID_NETWORK_STATE is returned

   NOTE: Any other suitable exception is accepted.

Test UI_38

Summary: **IpUICall**, sendInfoAndCollectReq methods, P_ILLEGAL_ID
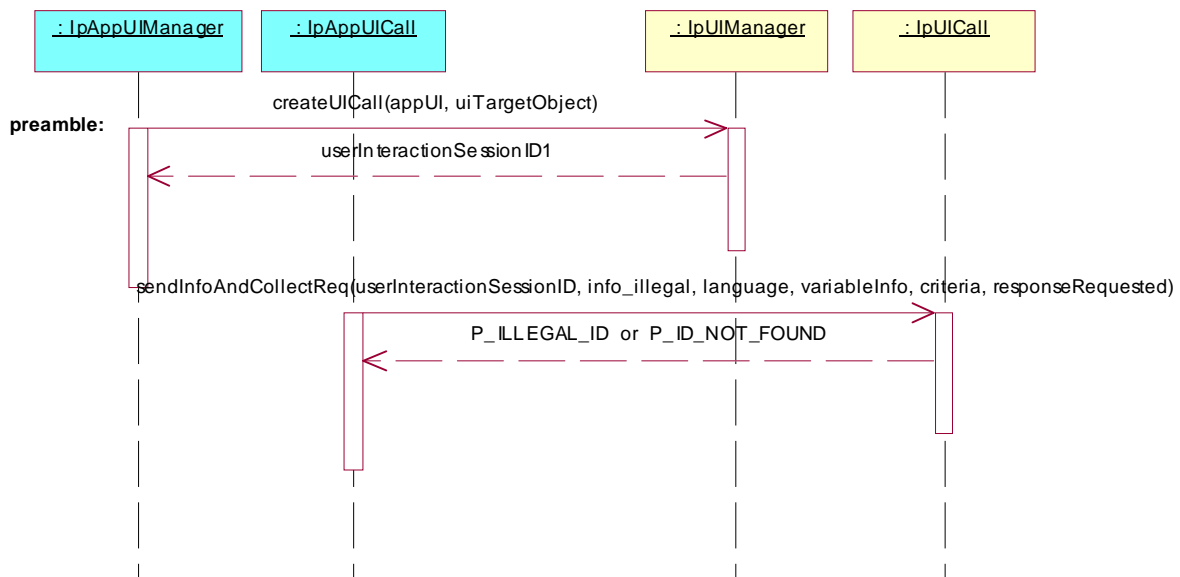
Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in the following methods to identify this User Interaction object.

Selection: sendInfoAndCollectReq method supported - PICS item: [2] UIC2

Test Sequence:

1. Method call **sendInfoAndCollectReq()**
   Parameters: userInteractionSessionID given in preamble, illegal info, language, variableInfo, criteria, responseRequested
   Check: P_ILLEGAL_ID or P_ID_NOT_FOUND is returned

NOTE: Any other suitable exception is accepted.

Test UI_39

Summary: **IpUICall**, sendInfoAndCollectReq methods, P_INVALID_CRITERIA

Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

Selection: sendInfoAndCollectReq method supported - PICS item: [2] UIC2
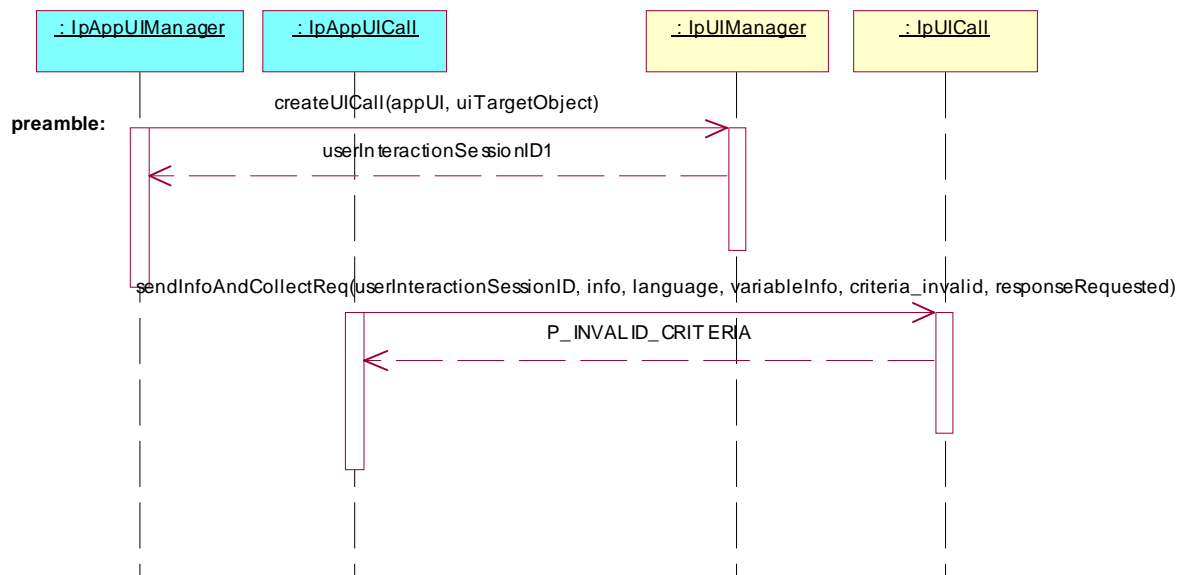
Test Sequence:

1.   Method call **sendInfoAndCollectReq()**
     Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, invalid criteria,
                    responseRequested.
     Check: P_INVALID_CRITERIA is returned

NOTE: Any other suitable exception is accepted.

Test UI_40

Summary: **IpUICall**, sendInfoAndCollectReq methods, P_ILLEGAL_RANGE

Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in the following methods to identify this User Interaction object.

Selection: sendInfoAndCollectReq method supported - PICS item: [2] UIC2
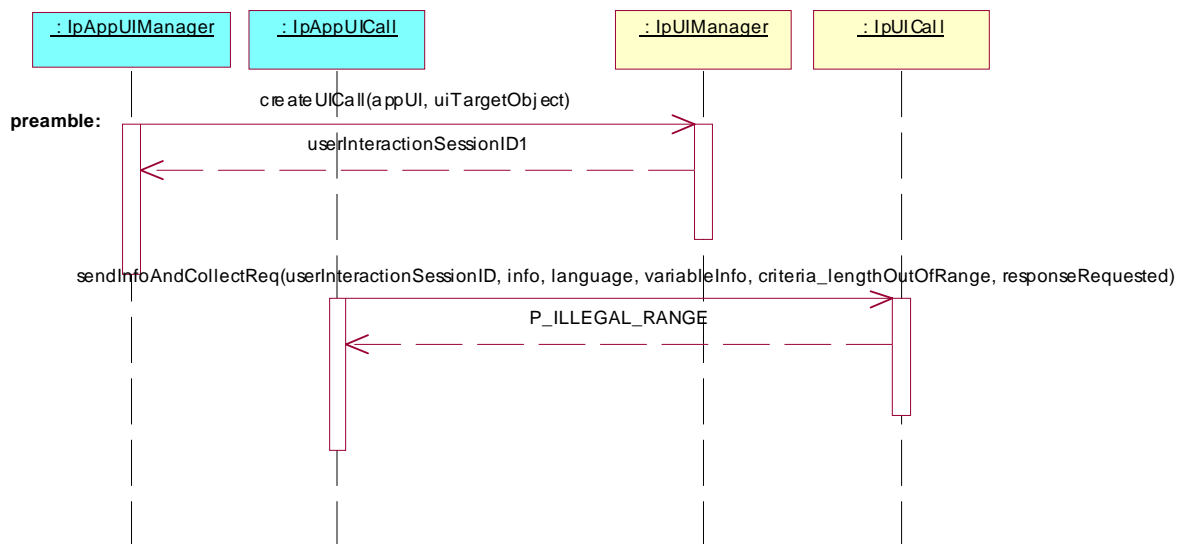
Test Sequence:

1. Method call **sendInfoAndCollectReq()**
    Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, criteria with collection length out of range, responseRequested.
    Check: P_ILLEGAL_RANGE is returned

NOTE: Any other suitable exception is accepted.

Test UI_41

Summary: **IpUICall**, sendInfoAndCollectReq methods, P_INVALID_COLLECTION_CRITERIA

Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

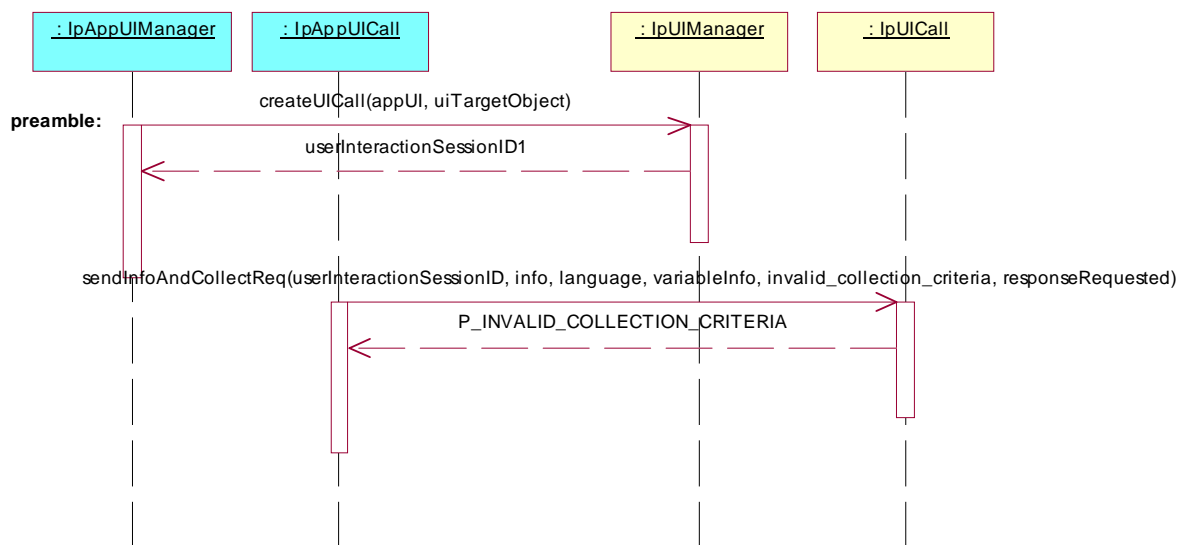Selection: sendInfoAndCollectReq method supported - PICS item: [2] UIC2

Test Sequence:

1. Method call **sendInfoAndCollectReq()**
    Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, invalid collection
                    criteria, responseRequested.
    Check: P_INVALID_COLLECTION_CRITERIA is returned

NOTE: Any other suitable exception is accepted.

Test UI_42

Summary: **IpUICall**, recordMessageReq methods, P_INVALID_SESSION_ID

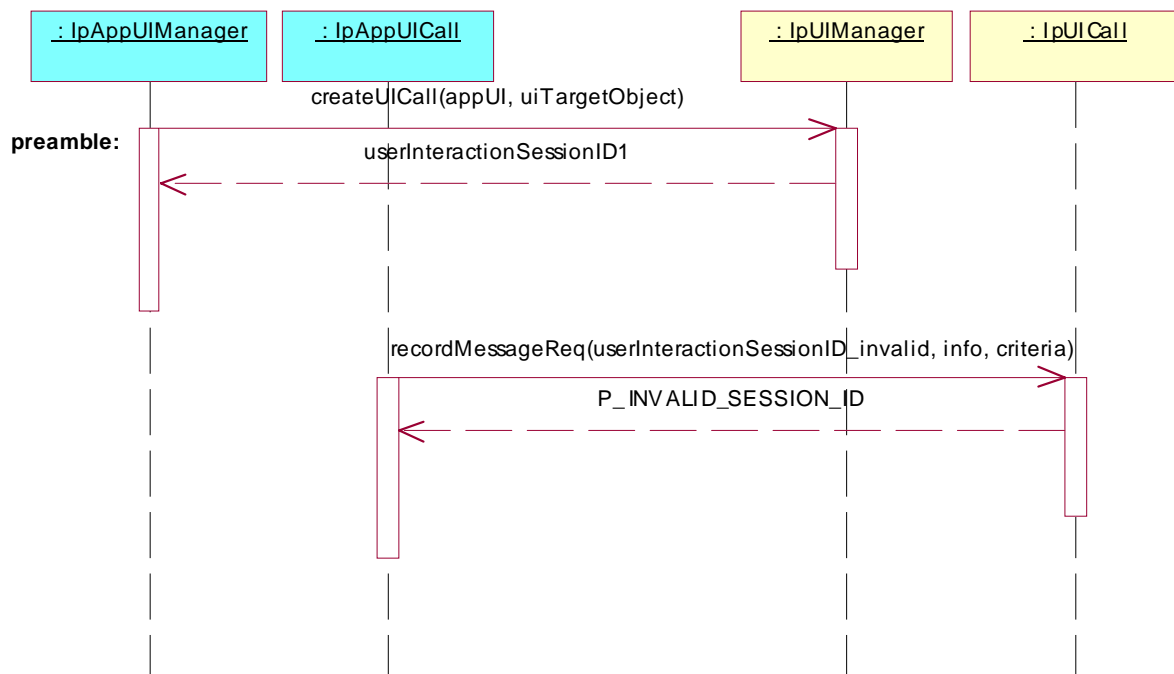Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

Selection: recordMessageReq method supported - PICS item: [2] UIC4

Test Sequence:

1. Method call **recordMessageReq()**
    Parameters: invalid userInteractionSessionID, info, criteria,
    Check: P_INVALID_SESSION_ID is returned

    NOTE: Any other suitable exception is accepted.

Test UI_43

Summary: **IpUICall**, recordMessageReq methods, P_INVALID_NETWORK_STATE

Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

Condition: the underlying protocol cannot support the method call (e.g. incompatible Call state)

Selection: recordMessageReq method supported - PICS item: [2] UIC4
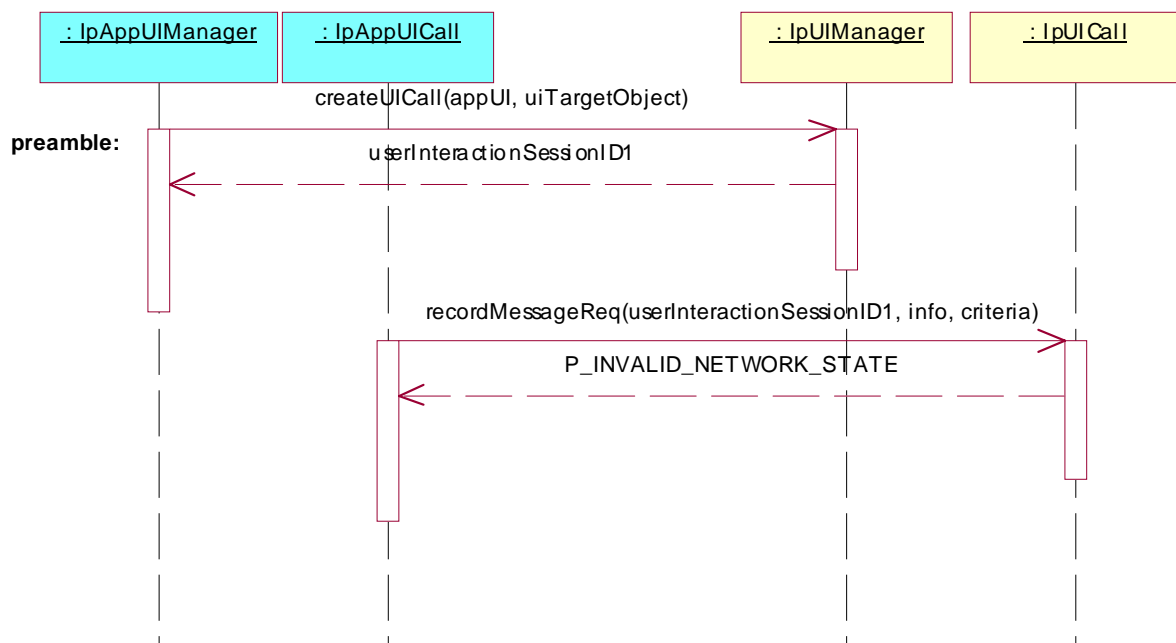
Test Sequence:

1. Method call **recordMessageReq()**
    Parameters: userInteractionSessionID given in preamble, info, criteria,
    Check: P_INVALID_NETWORK_STATE is returned

NOTE: Any other suitable exception is accepted.

Test UI_44

Summary: **IpUICall**, recordMessageReq methods, P_ILLEGAL_ID

Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

Selection: recordMessageReq method supported - PICS item: [2] UIC4
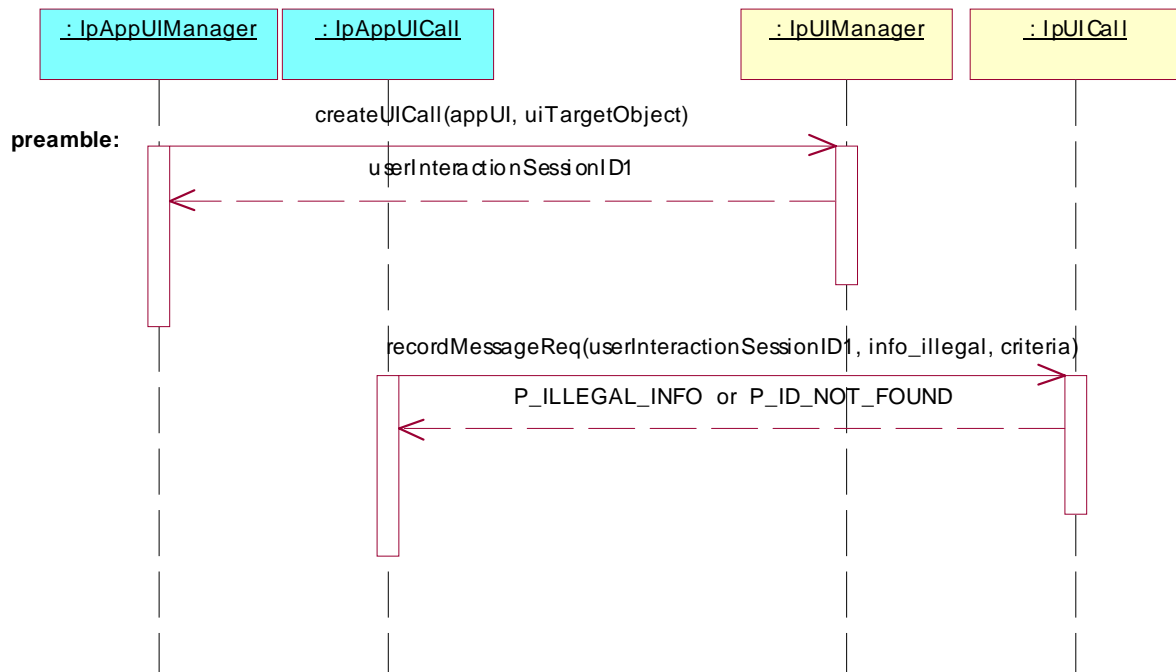
Test Sequence:

1.   Method call **recordMessageReq()**
     Parameters: userInteractionSessionID given in preamble, illegal info, criteria,
     Check: P_ILLEGAL_ID or P_ID_NOT_FOUND is returned

     NOTE: Any other suitable exception is accepted.

Test UI_45

Summary: **IpUICall**, recordMessageReq methods, P_INVALID_CRITERIA

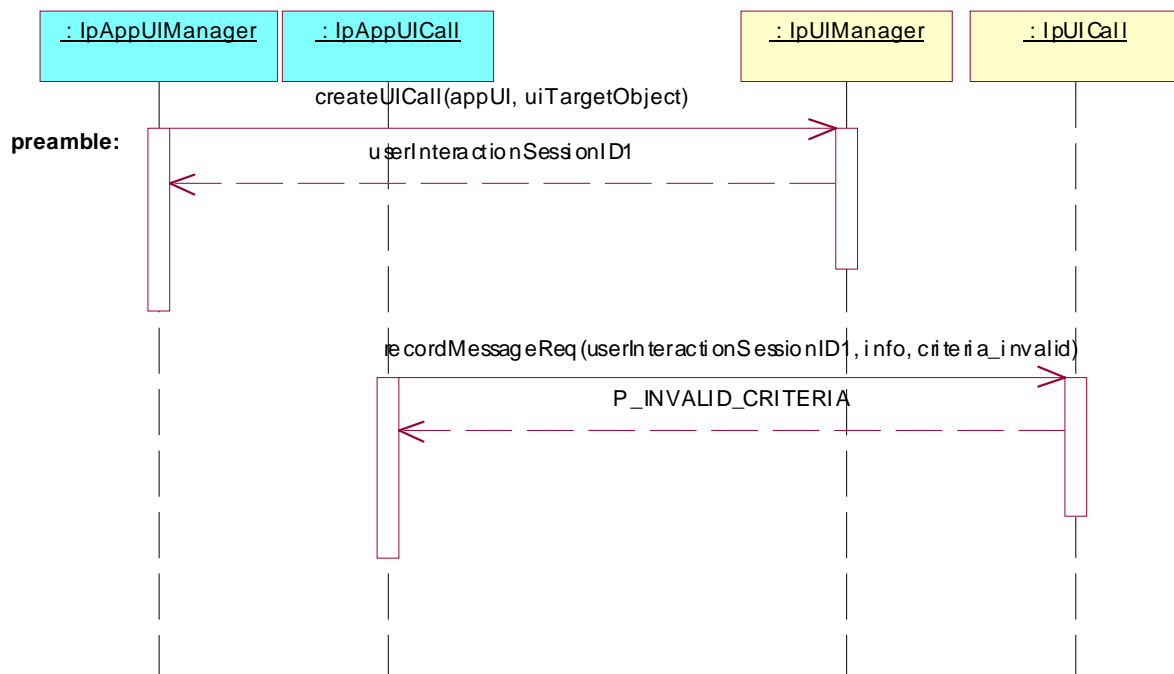Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
the following methods to identify this User Interaction object.

Selection: recordMessageReq method supported - PICS item: [2] UIC4

Test Sequence:

1. Method call **recordMessageReq()**
   Parameters: userInteractionSessionID given in preamble, info, invalid criteria,
   Check: P_INVALID_CRITERIA is returned

   NOTE: Any other suitable exception is accepted.

Test UI_46

Summary: **IpUICall**, deleteMessageReq, P_INVALID_SESSION_ID

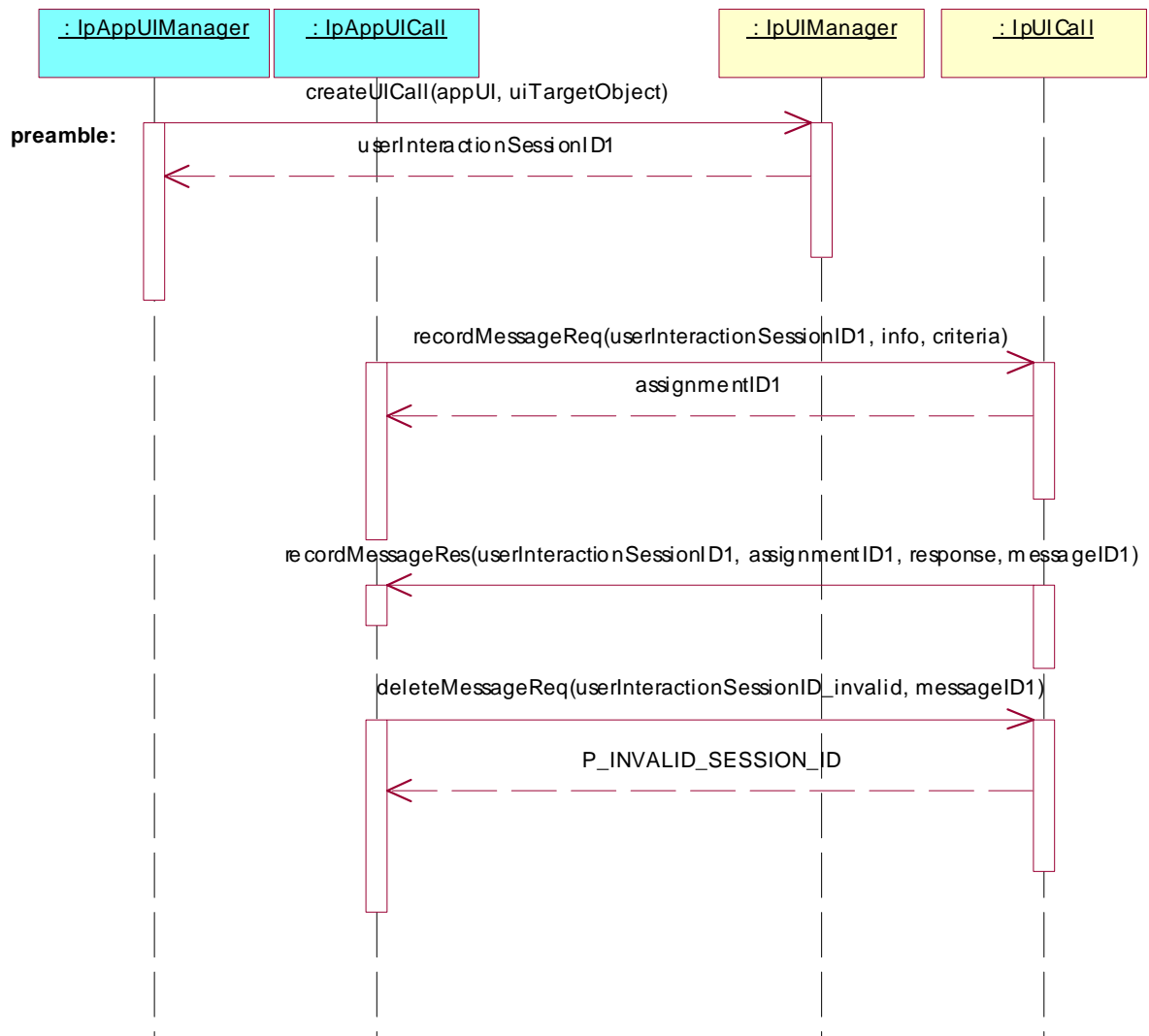Reference: ES 201 915-5 [1], clauses 8.5 and 8.6

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in the following methods to identify this User Interaction object.

Selection: recordMessageReq and deleteMessageReq methods supported - PICS item: [2] UIC4 AND UIC5

Test Sequence:

1.   Method call **recordMessageReq()**
     Parameters: userInteractionSessionID given in preamble, info, criteria.
     Check: valid value of TpAssignmentID is returned

2.   Triggered action: cause IUT to call **recordMessageRes()** method on the tester's (Application) **IpAppUICall** interface.
     Check: valid value of userInteractionSessionID (given in preamble), assignmentID (given in 1.), response and messageID is transmitted,
     Return: no parameter

3.   Method call **deleteMessageReq()**
     Parameters: invalid userInteractionSessionID, messageID returned in 2.
     Check: P_INVALID_SESSION_ID is returned

NOTE: Any other suitable exception is accepted.

Test UI_47

Summary: **IpUICall**, deleteMessageReq, P_ILLEGAL_ID

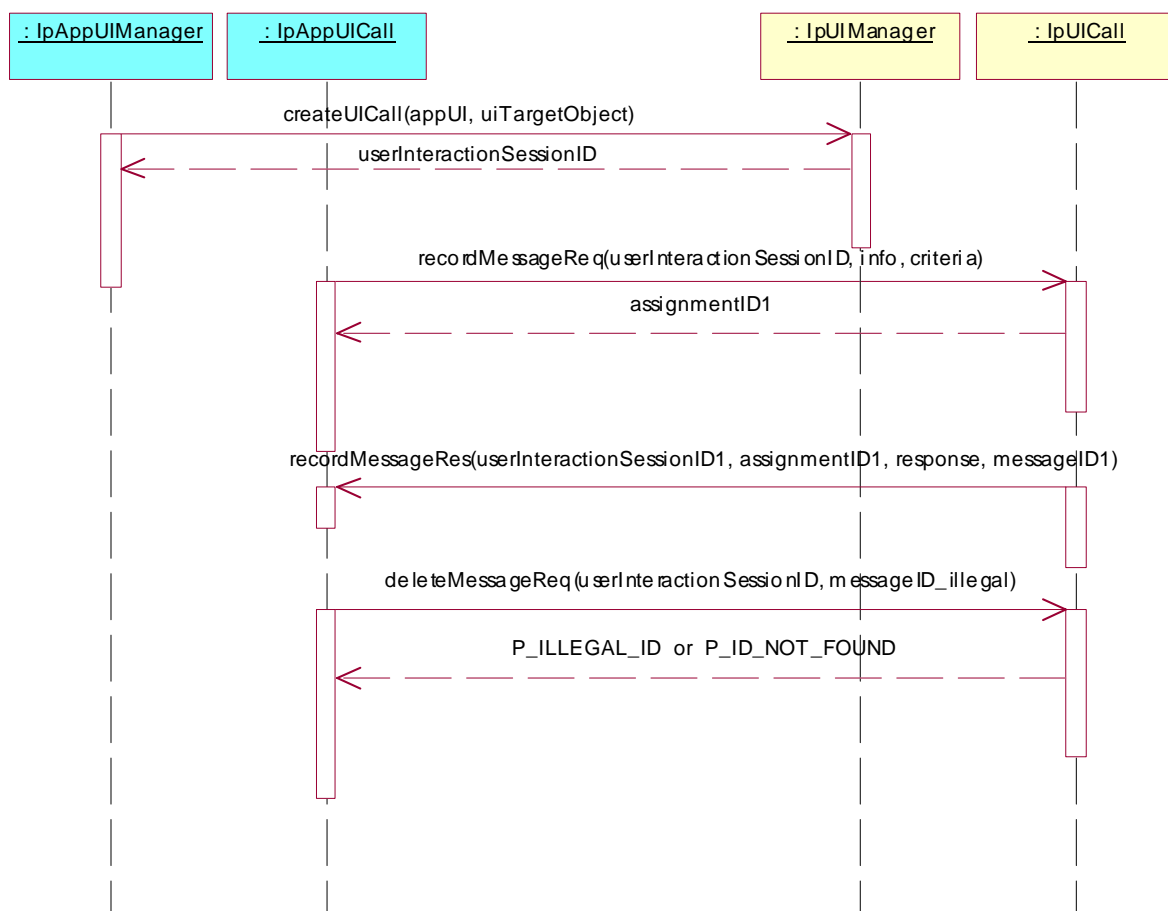Reference: ES 201 915-5 [1], clauses 8.5 and 8.6

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in the following methods to identify this User Interaction object.

Selection: recordMessageReq and deleteMessageReq methods supported - PICS item: [2] UIC4 AND UIC5

Test Sequence:

1.   Method call **recordMessageReq()**
     Parameters: userInteractionSessionID given in preamble, info, criteria.
     Check: valid value of TpAssignmentID is returned

2.   Triggered action: cause IUT to call **recordMessageRes()** method on the tester's (Application) **IpAppUICall** interface.
     Check: valid value of userInteractionSessionID (given in preamble), assignmentID (given in 1.), response and messageID is transmitted,
     Return: no parameter

3.   Method call **deleteMessageReq()**
     Parameters: userInteractionSessionID given in preamble, illegal messageID.
     Check: P_ILLEGAL_ID or P_ID_NOT_FOUND is returned

NOTE: Any other suitable exception is accepted.

Test UI_48

Summary: **IpUICall**, abortActionReq method, P_INVALID_SESSION_ID
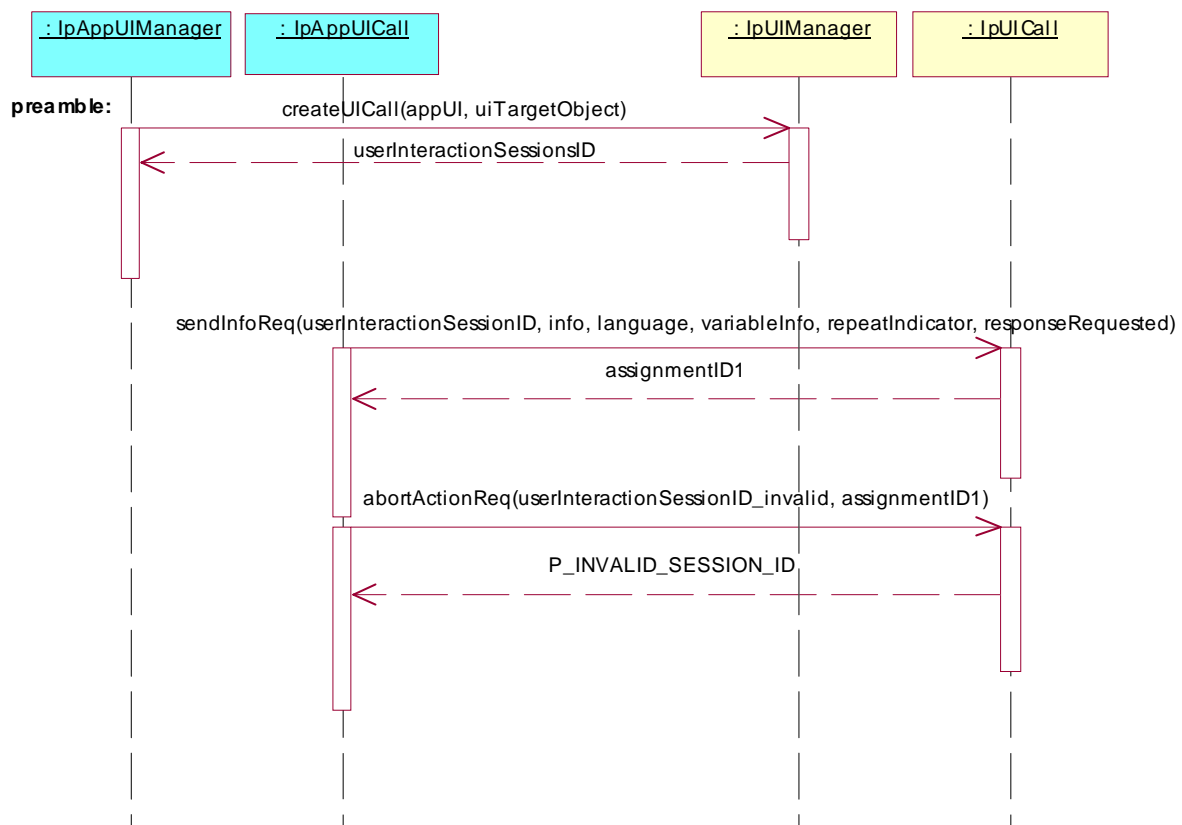
Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in the following methods to identify this User Interaction object.

Selection: sendInfoReq and abortActionReq methods supported - PICS item: [2] UIC1 AND UIC6

Test Sequence:

1. Method call **sendInfoReq()**
   Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, repeatIndicator, responseRequested.
   Check: valid value of TpAssignmentID is returned

2. Method call **abortActionReq()**
   Parameters: invalid userInteractionSessionID, assignmentID returned in 1.
   Check: P_INVALID_SESSION_ID is returned.

NOTE: Any other suitable exception is accepted.

Test UI_49

Summary: **IpUICall**, abortActionReq method, P_INVALID_ASSIGNMENT_ID

Reference: ES 201 915-5 [1], clause 8.5
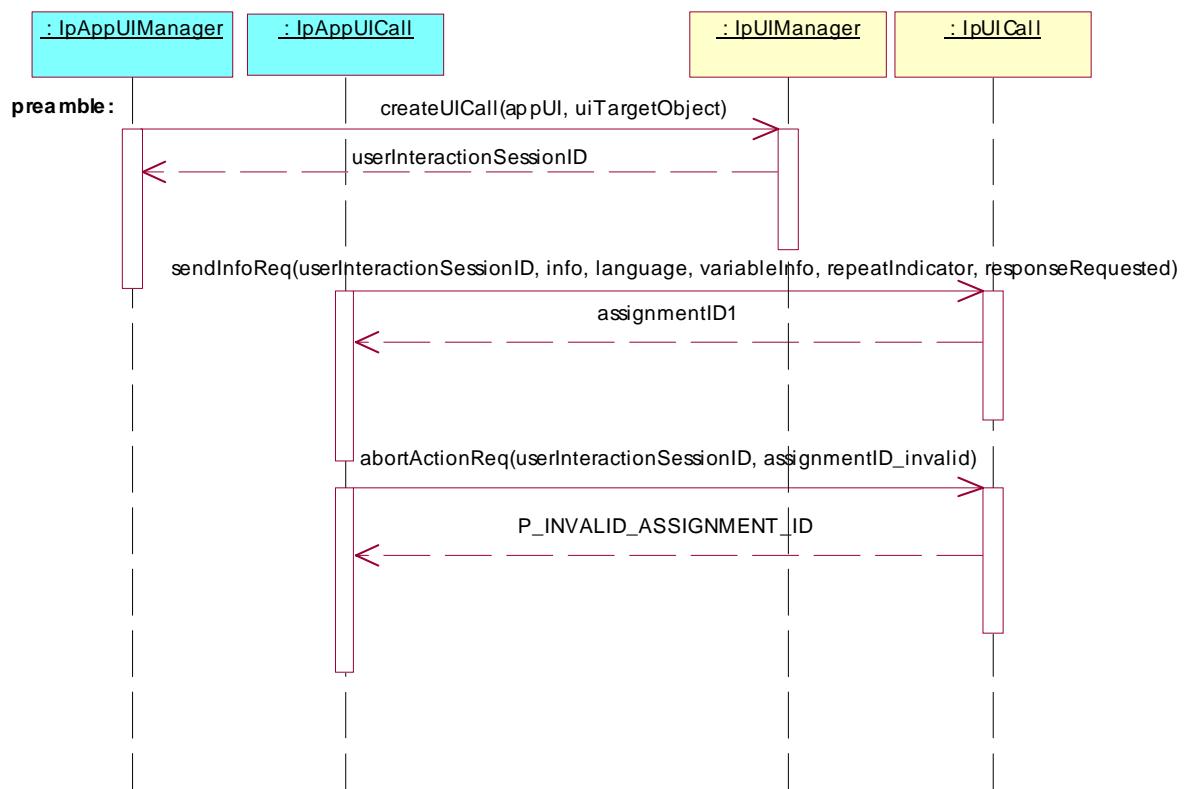
Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                   IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                   the following methods to identify this User Interaction object.

Selection: sendInfoReq and abortActionReq methods supported - PICS item: [2] UIC1 AND UIC6

Test Sequence:

   1.   Method call **sendInfoReq()**
        Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, repeatIndicator,
                       responseRequested.
        Check: valid value of TpAssignmentID is returned

   2.   Method call **abortActionReq()**
        Parameters: userInteractionSessionID given in preamble, invalid assignmentID.
        Check: P_INVALID_ASSIGNMENT_ID is returned.

   NOTE: Any other suitable exception is accepted.

Test UI_50

Summary: **IpUICall**, abortActionReq method, P_INVALID_SESSION_ID

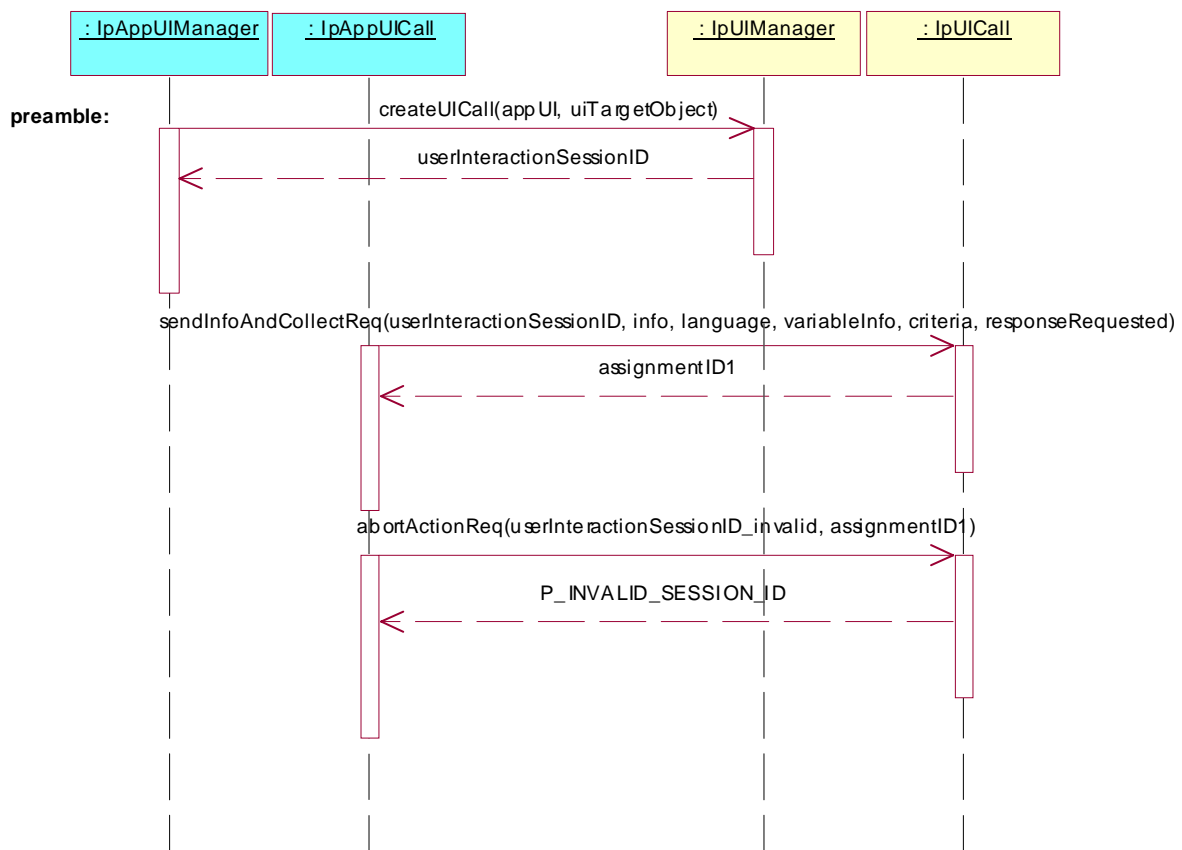Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

Selection: sendInfoAndCollectReq and abortActionReq methods supported - PICS item: [2] UIC2 AND UIC6

Test Sequence:

   1. Method call **sendInfoAndCollectReq()**
       Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, criteria,
                       responseRequested.
       Check: valid value of TpAssignmentID is returned

   2. Method call **abortActionReq()**
       Parameters: invalid userInteractionSessionID, assignmentID returned in 1.
       Check: P_INVALID_SESSION_ID is returned.

   NOTE: Any other suitable exception is accepted.

Test UI_51

Summary: **IpUICall**, abortActionReq method, P_INVALID_ASSIGNMENT_ID

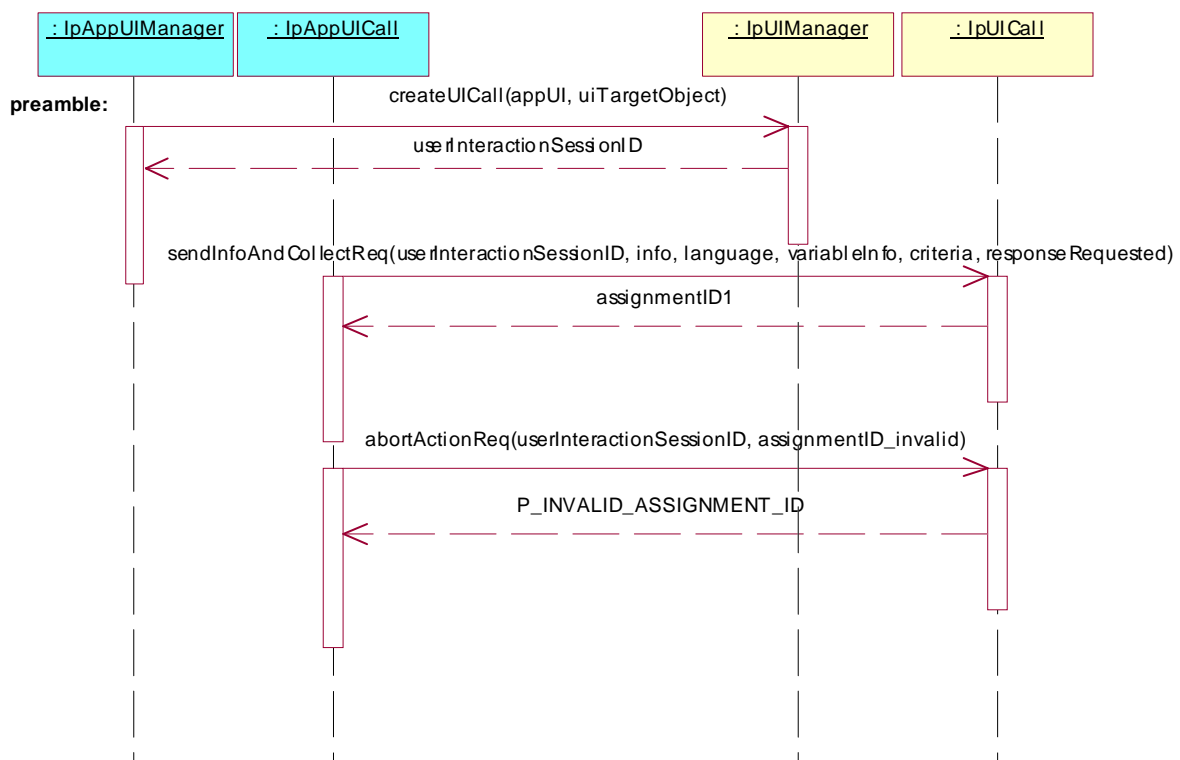Reference: ES 201 915-5 [1], clause 8.5

Preamble: A User Interaction object, created either with the createUI or the createUICall method on the
                IpUIManager interface or reported in a reportNotification method exists. The given sessionID is used in
                the following methods to identify this User Interaction object.

Selection: sendInfoAndCollectReq and abortActionReq methods supported - PICS item: [2] UIC2 AND UIC6

Test Sequence:

1.   Method call **sendInfoAndCollectReq()**
      Parameters: userInteractionSessionID given in preamble, info, language, variableInfo, criteria,
                      responseRequested.
      Check: valid value of TpAssignmentID is returned

2.   Method call **abortActionReq()**
      Parameters: userInteractionSessionID given in preamble, invalid assignmentID.
      Check: P_INVALID_ASSIGNMENT_ID is returned.

NOTE: Any other suitable exception is accepted.

History

Document history			
V1.1.1	June 2003	Membership Approval Procedure	MV 20030801: 2003-06-03 to 2003-08-01