

PGED-ACTA-STE011-ACARIS

Modulo “Servizi Applicativi”

ACTA Archive Interoperability Services

Specifica dei BackOffice Services

Versione <5.0>

VERIFICHE E APPROVAZIONI

VERSIONE	REDAZIONE		CONTROLLO APPROVAZIONE		AUTORIZZAZIONE EMISSIONE	
	NOME	DATA	NOME	DATA	NOME	DATA
1.0	GARBOLINO MARANDO	12/11/2010			CERONI	12/11/2010
2.0	MARANDO	25/03/2011			CERONI	28/03/2011
3.0	MARANDO	29/04/2011			CERONI	02/05/2011
4.0	MARANDO	05/06/2013			CERONI	06/06/2013
5.0	COERO-BORGA	02/08/2022			MARENCHINO	02/08/2022

STATO DELLE VARIAZIONI

VERSIONE	PARAGRAFO O PAGINA	DESCRIZIONE DELLA VARIAZIONE
V01	Tutto il documento	Versione iniziale del documento
V02	Descrizione dell'interfaccia del servizio	Approfondimento su getDescendants e getObjectParents. Scorporo della funzionalità di navigazione dai due servizi precedenti relativamente alle gerarchie tra strutture e messa a disposizione di un servizio ad hoc
V03	Vari	<ul style="list-style-type: none"> • Stabilite le regole per l'accesso agli oggetti BKO in operazioni di lettura (navigazione, ricerca, dettaglio) • Modificati ed integrati esempi e diagrammi di riferimento • Previsti percorsi di navigazione da impostare come parametro del servizio getDescendants • Precisata interfaccia del servizio getProperties • Inserito servizio getPropertiesMassive
V04	Paragrafo 3.13	Inserito servizio getPrincipalExt
V05		Aggiornato la descrizione delle operazioni getQueryableObjects e getQueryableObjectMetadata Aggiunto riferimenti al wsdl e xsd dell'ambiente di test

INDICE

Introduzione.....	6
Scopo del documento	6
Riferimenti	6
Servizio BackOffice Service.....	7
Obiettivi	7
Modalità di richiamo	7
Riferimento Oggetti Domain Model ACARIS	8
Regole per l'accesso in sola lettura agli oggetti di BKO	8
Descrizione dell'interfaccia del servizio	9
Operazione getDettaglioAOO	9
Descrizione dell'operazione.....	9
Interfaccia di richiamo (input)	9
Interfaccia di output	9
Logica di business	9
Operazione getDettaglioStruttura	9
Descrizione dell'operazione.....	9
Interfaccia di richiamo (input)	9
Interfaccia di output	10
Logica di business	10
Operazione getPrincipal	10
Descrizione dell'operazione.....	10
Interfaccia di richiamo (input)	10
Interfaccia di output	11
Logica di Business	11
Operazione getQueryableObjects.....	11
Descrizione dell'operazione.....	11
Interfaccia di richiamo (input)	11
Interfaccia di output	12
Logica di Business	12
Operazione getQueryableObjectMetadata	12
Descrizione dell'operazione.....	12
Interfaccia di richiamo (input)	12
Interfaccia di output	13
Operazione query	14
Descrizione dell'operazione.....	14
Interfaccia di richiamo (input)	14
Interfaccia di output	14

Logiche di Business	14
Caratteristiche comuni alle operazioni di navigazione e ricerca	15
Esposizione delle informazioni relative al “parent” dell’unità organizzativa	15
Operazione getDescendants.....	16
Descrizione dell’operazione.....	16
Interfaccia di richiamo (input)	16
Interfaccia di output	16
Percorsi di navigazione	16
Navigazione attraverso l’AOO.....	17
Navigazione dall’Ente alle Strutture	21
Navigazione dall’Ente agli Utenti	21
Operazione getObjectParents	21
Descrizione dell’operazione.....	21
Interfaccia di richiamo (input)	21
Interfaccia di output	22
Operazione navigaGerarchia	24
Descrizione dell’operazione.....	24
Interfaccia di richiamo (input)	24
Interfaccia di output	24
Logiche di Business	24
<u>Considerazioni sul parametro role</u>	24
Esposizione delle informazioni relative alla struttura “parent” nelle gerarchie.....	24
Operazione getProperties	25
Descrizione dell’operazione.....	25
Interfaccia di richiamo (input)	25
Interfaccia di output	25
Logiche di Business	25
Operazione getPropertiesMassive.....	26
Descrizione dell’operazione.....	26
Interfaccia di richiamo (input)	26
Interfaccia di output	26
Logiche di Business	26
Operazione getPrincipalExt	26
Descrizione dell’operazione.....	26
Interfaccia di richiamo (input)	27
Interfaccia di output	27
Eccezioni specifiche	27
Logica di Business	27
Governance	28
Requisiti di sicurezza	28

Figura 1 - Gerarchia degli oggetti del modulo BackOffice.....	7
Figura 2 - tipologie oggetti getDescendants	18
Figura 3 - getDescendants con esempi percorsi di navigazione	19
Figura 4 - getDescendants con evidenziazione oggetti non restituibili.....	20
Figura 5 - getObjectParents con esempio percorsi di navigazione	23

Introduzione

Scopo del documento

Questo documento specifica tutte le operazioni possibili sugli oggetti del modulo funzionale Acaris BackOffice.

Riferimenti

- [A1] PGED-ACTA-SRS-v14_ Requisiti funzionali.doc
- [A2] PGED-ACTA-SRS-ACARIS-V01.doc
- [A3] PGED-ACTA-STE00-ACARIS-V01-DomainModel.doc
- [A4] PGED-ACTA-UC-ACARISDOC01-V01_Crea_DocumentoArchivistico.doc
- [A5]
- [A6]
- [A7] PGED-ACTA-UC-ACARISREL01-V01_Crea_DocumentComposition.doc
- [A8] PGED-ACTA-UC-ACARISFOLD02-V01_Crea_DocFisico.doc
- [A9] PGED-ACTA-UC-ACARISFOLD003-V01_Crea_fascicolo.doc
- [A10] PGED-ACTA-Architettura_V02.doc
- [A11] PGED-ACTA-UC-ACARISASS01-V01_Crea_Documento_Classificato.doc
- [A12] PGED-ACTA-UC-ACARISASS02-V01_Crea_Documento_Elettronico.doc
- [A13] backofficeWS?xsd=ACARIS-BackOfficeMessaging.xsd
- [A14] backofficeWS?xsd=ACARIS-BackOffice.xsd
- [A15] backofficeWS?xsd=ACARIS-Common.xsd

Servizio *BackOffice Service*

Obiettivi

L'insieme delle operazioni possibili nell'ambito dei BackOffice Services permette la gestione di oggetti di tipo BackOffice derivanti direttamente da Properties.

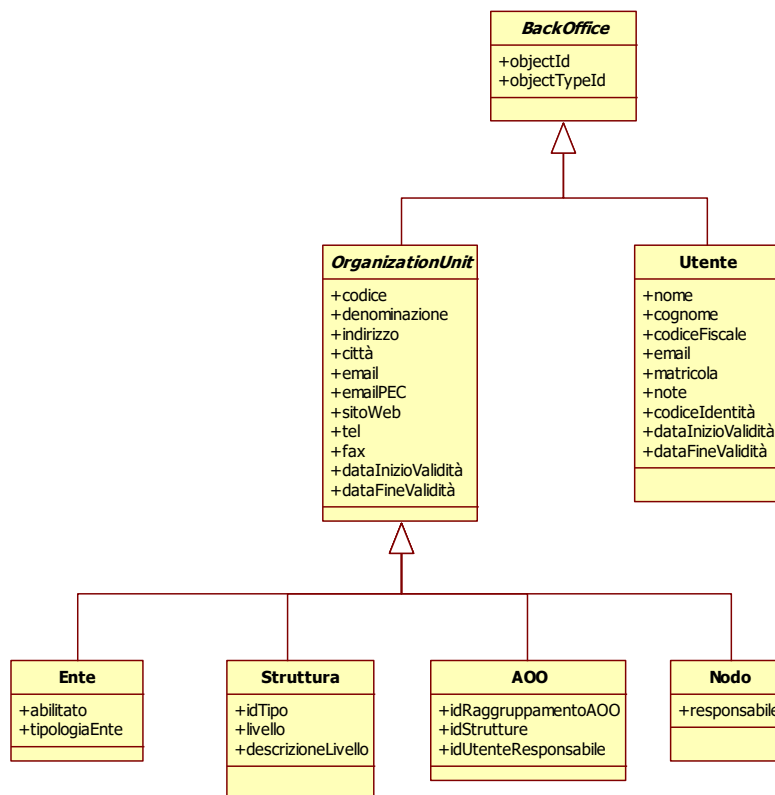


Figura 1 - Gerarchia degli oggetti del modulo BackOffice

Sono previste generiche operazioni di navigazione del repository e altre più specifiche come ad esempio il reperimento dei principal.

Modalità di richiamo

Per quanto riguarda il richiamo dei servizi avviene in modalità sincrona.

Essi sono esposti tramite il servizio di infrastruttura CSI (PA/PD) e Web Services (SOAP), si rimanda al documento di architettura [A10].

Riferimento Oggetti Domain Model ACARIS

Nel seguito i riferimenti agli oggetti del domain data model degli ACTA Services, come descritto in [A3], sono effettuati per mezzo dei nomi definiti negli schema di riferimento ed in particolare:

- **complexType**

```
<xs:complexType name="ObjectTypeId">
  <xs:complexContent>
    <xs:sequence>
      <xs:element name="PropertyId" type="PropertyType"/>
      ...
    </xs:sequence>
  </xs:complexContent>
</xs:complexType>
```
- **simpleType**

```
<xs:simpleType name="PropertyType">
  ...
</xs:simpleType>
```

Il nome utilizzato è quello specificato per l'attributo *name* dei tipi complessi o semplici definiti negli schema ACARIS-BackOffice.xsd o ACARIS-BackOfficeMessaging.xsd. Può trattarsi a seconda dei casi del nome della classe dell'oggetto del modello (attributo ObjectTypeId) oppure il nome della property (attributo PropertyId) oppure il tipo della property (attributo PropertyType).

Regole per l'accesso in sola lettura agli oggetti di BKO

Diversi servizi di BackOffice mettono a disposizione operazioni di sola lettura: sono compresi i servizi di navigazione, query, visualizzazione dettaglio oggetti.

Secondo le regole di Acta l'accesso agli oggetti dovrebbe essere consentito soltanto se vengono soddisfatte le regole di visibilità e/o profilazione e/o collocazione.

L'accesso a servizi di tali oggetti, però, richiede che le operazioni di sola lettura vengano consentite in forma libera: si stabilisce, quindi, che per Acaris non operano le regole suddette e l'accesso è consentito liberamente conformemente a quanto di seguito specificato.

L'accesso agli oggetti di tipo Ente, GruppoAOO, AOO, Struttura, Nodo, Utente, Identità, Profilo, Funzione è libero. Si richiede solo che il richiedente operi all'interno dell'ente di appartenenza: non è consentito l'accesso agli oggetti di un ente diverso.

L'amministratore di sistema però può accedere agli oggetti di tutti gli enti.

Non sono previste, quindi, limitazioni a particolari campi degli oggetti (es: codice fiscale dell'utente...), limitazioni legate alla posizione dell'identità nell'organigramma o limitazioni legate a profili particolari.

Descrizione dell'interfaccia del servizio

Operazione getDettaglioAOO

Descrizione dell'operazione

Restituisce le informazioni di dettaglio di una AOO specificata.

Interfaccia di richiamo (input)

- IdAOO
- repositoryId

Interfaccia di output

Collection properties dell' AOO:

- ✓ idAOO
- ✓ codice
- ✓ denominazione
- ✓ indirizzo
- ✓ città
- ✓ email
- ✓ emailPEC
- ✓ sitoWeb
- ✓ tel
- ✓ fax

Logica di business

Le logiche di business sono descritte nel caso d'uso ACARISMAN002 rif. [A05]

Operazione getDettaglioStruttura

Descrizione dell'operazione

Restituisce le informazioni di dettaglio di una Struttura specificata.

Interfaccia di richiamo (input)

- IdStruttura
- repositoryId

Interfaccia di output

Collection properties della Struttura:

- ✓ idStruttura
- ✓ codice
- ✓ denominazione
- ✓ indirizzo
- ✓ città
- ✓ email
- ✓ emailPEC
- ✓ sitoWeb
- ✓ tel
- ✓ fax

Logica di business

Le logiche di business sono descritte nel caso d'uso ACARISMAN003 rif. [A06]

Operazione getPrincipal

Descrizione dell'operazione

Questa operazione permette di ottenere l'elenco dei 'principal' che possono operare sul repository dato il codice fiscale di un utente autenticato dal sistema fruitore.

Interfaccia di richiamo (input)

- IdRepository: Repository Id
- Codice fiscale: Codice fiscale dell' attore autenticato
- IdAoo : Identificativo AOO di lavoro (opzionale)
- IdStruttura : Identificativo Struttura di lavoro (opzionale)
- IdNodo : Identificativo Nodo di lavoro (opzionale)

Codice Fiscale

E' il codice fiscale di un utente presente su ACTA ed abilitato ad operare sul repository (Ente).

IdAOO

Nella chiamata all' operazione se specificata l' AOO il sistema ritorna solo i principal abilitati ad operare sull'AOO indicata.

IdStruttura

Nella chiamata all' operazione se specificata la Struttura il sistema ritorna solo i principal abilitati ad operare sulla Struttura indicata, se specificata la struttura è obbligatorio che sia specificato anche l'IdAOO.

IdNodo

Nella chiamata all' operazione se specificato il Nodo il sistema ritorna solo i principal abilitati ad operare sul Nodo indicato, se specificato il nodo è obbligatorio che sia specificato anche l'IdAOO e l'IdStruttura.

Interfaccia di output

- ✓ Elenco di oggetti contenente:
 - IdPrincipal
 - IdAoo
 - IdStruttura
 - IdNodo

Logica di Business

Le logiche di business sono descritte nel caso d'uso ACARISMAN001 rif. [A04]

Operazione getQueryableObjects

Descrizione dell'operazione

Questa operazione permette di ottenere l'elenco degli oggetti di tipo Queryable che possono essere utilizzati nelle operazioni di query come target.

L'operazione è così definita (vedere [A13]):

```
<xs:element name="getQueryableObjects">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="1" minOccurs="1" name="repositoryId"
        type="common:ObjectIdType"/>
      <xs:element maxOccurs="1" minOccurs="1" name="service" type=
        ="tns:enumServiceType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Può trattarsi di entità fisiche oppure di viste create dinamicamente dal sistema.
L'elenco è specifico per l'ente di interesse.

Interfaccia di richiamo (input)

- idRepository: Repository Id
- service: enumServiceType

idRepository

- Identificativo del repository (Ente) di interesse

service (vedere [A14])

```
<xs:simpleType name="enumServiceType">
  <xs:restriction base="common:string">
    <xs:enumeration value="Archive"/>
    <xs:enumeration value="BackOffice"/>
    <xs:enumeration value="Management"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:enumeration value="OfficialBook"/>
<xs:enumeration value="Sms"/>
<xs:enumeration value="SubjectRegistry"/>
</xs:restriction>
</xs:simpleType>
```

Interfaccia di output

- ✓ Elenco di oggetti di tipo Queryable: sostanzialmente si tratta di un elenco di stringhe.

Logica di Business

Operazione getQueryableObjectMetadata

Descrizione dell'operazione

Questa operazione permette di ottenere l'elenco dei metadati associati ad un oggetto di tipo Queryable utilizzabile in un'operazione di query.

L'elenco è specifico per l'ente di interesse

L'operazione è così definita (vedere [A13]):

`ObjectMetadataType[] getQueryableObjectMetadata(ObjectIdType repositoryId,`
`QueryableObjectType queryableObject)`

```
<xs:element name="getQueryableObjectMetadata">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="1" minOccurs="1" name="repositoryId"
        type="common:ObjectIdType"/>
      <xs:element maxOccurs="1" minOccurs="1" name="queryableObject"
        type="common:QueryableObjectType"/>
      <xs:element maxOccurs="1" minOccurs="1" name="operation"
        type="common:enumPropertyFilterOperation"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Ogni ObjectMetadata contiene informazioni che consentono di stabilire se lo stesso può essere utilizzato come oggetto di ritorno di una query oppure come criterio di filtro.

Quando utilizzato come criterio di filtro si possono utilizzare le informazioni relative al tipo di dato ed agli operatori di confronto.

E' inoltre possibile sapere se l'oggetto può essere aggiornato tramite l'operazione updateProperties.

Interfaccia di richiamo (input)

- objectIdType: repositoryId
- QueryableObject: queryableObject
- enumPropertyFilterOperation: operation

queryableObject

Identificatore ottenuto tramite precedente operazione getQueryableObjects (vedere [A15])

```
<xs:complexType name="QueryableObjectType">
  <xs:sequence>
    <xs:element maxOccurs="1" minOccurs="1" name="object" type="tns:string"/>
  </xs:sequence>
</xs:complexType>
```

operation (vedere [A15])

```
<xs:simpleType name="enumPropertyFilterOperation">
  <xs:restriction base="tns:string">
    <xs:enumeration value="none"/>
    <xs:enumeration value="all"/>
    <xs:enumeration value="query"/>
    <xs:enumeration value="getProperties"/>
    <xs:enumeration value="getDescendants"/>
    <xs:enumeration value="getChildren"/>
    <xs:enumeration value="getFolderParent"/>
    <xs:enumeration value="getObjectParents"/>
  </xs:restriction>
</xs:simpleType>
```

Interfaccia di output

- ✓ Elenco di oggetti di tipo ObjectMetadataType: l'oggetto è così definito (vedere [A15]):

```
<xs:complexType name="ObjectMetadataType">
  <xs:sequence>
    <xs:element maxOccurs="1" minOccurs="1" name="queryName" type="tns:QueryNameType"/>
    <xs:element maxOccurs="1" minOccurs="1" name="selectable" type="tns:boolean"/>
    <xs:element maxOccurs="1" minOccurs="1" name="queryable" type="tns:boolean"/>
    <xs:element maxOccurs="unbounded" minOccurs="1" name="operators" type="tns:string"/>
    <xs:element maxOccurs="1" minOccurs="1" name="dataType" type="tns:string"/>
    <xs:element maxOccurs="1" minOccurs="1" name="updatable" type="tns:boolean"/>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="propertyFilterConfigurationInfo" type="tns:PropertyFilterConfigurationInfoType"/>
  </xs:sequence>
</xs:complexType>
```

Logica di Business

Operazione query

Descrizione dell'operazione

Consente di effettuare delle query sul modulo funzionale Acaris BackOffice.

Per un approfondimento si veda l'apposita sezione in allegato tecnico.

Si richiama quanto stabilito nel paragrafo "Regole per l'accesso in sola lettura agli oggetti di BKO" per l'individuazione delle regole di accesso da applicare nelle operazioni di navigazione (applicabili a query in quanto operazione assimilabile).

Interfaccia di richiamo (input)

- repositoryId
- principalId
- target
- filter
- criteria
- maxItems
- skipCount

I parametri del tipo RepositoryIdType e PrincipalIdType non presentano particolarità rispetto agli altri servizi.

QueryableObjectType target

Rappresenta l'entità interrogabile. Può essere un'entità fisica oppure una vista dinamica.

PropertyFilterType filter

Permette di specificare l'elenco delle properties che devono essere restituite per ogni oggetto individuato nel sistema.

QueryConditionType[] criteria

Elenco di criteri (in questa prima versione concatenate in AND) di selezione.

Integer maxItems

Integer skipCount

Vengono utilizzati per gestire la paginazione dei risultati.

Interfaccia di output

Logiche di Business

Per un approfondimento in merito a interfaccia di input, parametri, interfaccia di output si veda l'apposita sezione dell'allegato tecnico di Acta.

Caratteristiche comuni alle operazioni di navigazione e ricerca

Si richiama quanto stabilito nel paragrafo “Regole per l’accesso in sola lettura agli oggetti di BKO” per l’individuazione delle regole di accesso da applicare nelle operazioni di navigazione.

La navigazione parte da un nodo identificato come root che non viene ricompreso nell’insieme di risultati restituiti dal servizio.

La navigazione, in ascendenza o discendenza, dell’organigramma di BKO, può coinvolgere percorsi abbastanza complicati e difficilmente riconducibili ad un risultato di valore per gli utilizzatori. In considerazione di ciò, si decide di seguire un approccio che privilegi la semplicità di navigazione basata sull’utilizzo di percorsi lineari. Le strutture incontrate nel percorso di discendenza sono restituite senza investigare gli eventuali rapporti gerarchici tra le stesse: tali navigazioni costituiscono oggetto di un servizio separato.

Gli oggetti restituibili, avuto riguardo alla tipologia del nodo root ed alle caratteristiche del singolo servizio (si vedano, ad esempio, le regole previste per i percorsi di navigazione in getDescendants), sono dei seguenti tipi: Ente, GruppoAOO, AOO, Struttura, Nodo, Utente.

Tutte le operazioni coinvolte tengono conto della data passata come parametro per contestualizzare alla stessa l’individuazione delle occorrenze attive (valide).

Esposizione delle informazioni relative al “parent” dell’unità organizzativa

Le operazioni di navigazione possono restituire oggetti rintracciabili a diversi livelli dell’organigramma. Per consentire di ricostruire i legami tra le unità organizzative, è necessario che per ognuna di esse il sistema restituisca l’identificatore dell’unità padre.

Poiché alcune unità organizzative possono avere legami con diversi “parent” si rende necessaria una scelta tra le diverse opzioni. Il seguente schema indica le sole relazioni considerate:

Utente → Nodo → Struttura → AOO → GruppoAOO → Ente

Segue l’elenco di come devono essere valorizzati i “parent” per ogni Unità organizzativa tenendo conto anche dei percorsi di navigazione quando la stessa avviene tramite getDescendants:

Utente → “Nodo;” Nodo.codice (oppure “Ente;” Ente.codice nel caso di percorso di navigazione “dall’ente agli utenti” nel servizio getDescendants)

Nodo → “Struttura;” Struttura.codice

Struttura → “AOO;” AOO.codice (oppure “Ente;” Ente.codice nel caso di percorso di navigazione “dall’ente alle strutture” nel servizio getDescendants)

AOO → “GruppoAOO;” GruppoAOO.descrizione

GruppoAOO → “Ente;” Ente.codice

Esempi:

Utente.parent = “Nodo;100”

Nodo.parent = “Struttura;010”

Struttura.parent = “AOO;001”

AOO.parent = “GruppoAOO;GruppoAOO Didattica”

GruppoAOO.parent = “Ente;001”

Operazione getDescendants

Descrizione dell'operazione

Operazione di navigazione in profondità dell'organigramma.

Interfaccia di richiamo (input)

- ✓ **ObjectIdType repositoryId:** Repository e quindi ente di riferimento.
- ✓ **PrincipalIdType principalId:** Utente che richiede il servizio.
- ✓ **EnumBackOfficeNavigationPath path:** percorso di navigazione selezionato
- ✓ **ObjectIdType rootNodeId:** identificatore dell'unità organizzativa a partire dalla quale iniziare la navigazione.
- ✓ **date date:** serve per limitare la navigazione alle sole unità attive (valide) alla data passata come parametro.
- ✓ **integer depth:** Livello di profondità dell'operazione di navigazione.
- ✓ **PropertyFilterType filter:** permette di indicare tipologie di unità organizzative e proprietà da restituire
- ✓ **MaxItems :** determina il numero massimo di oggetti da restituire. Si rimanda a quanto stabilito per ciascun Ente in merito al numero massimo di oggetti restituibili.
- ✓ **skipCount :** Determina l'elemento da cui continuare nella restituzione degli oggetti trovati durante la navigazione del folder

Interfaccia di output

- ✓ **<Array> ObjectResults:** Elenco di nodi dell'organigramma.
 - **ObjectIdType objectId:** Nodo dell'albero dell'organigramma.
 - **<Array> PropertyType properties:** Proprietà valorizzate per il nodo.
- ✓ **hasMoreItems :** indica se ci sono ulteriori risultati quando viene utilizzato maxItems per la paginazione dei risultati

Il servizio permette di navigare in discendenza l'organigramma di BKO scegliendo un percorso di navigazione ed un nodo identificato come root (rootNodeId) . I risultati restituiti sono individuati a partire dal livello successivo a quello di root.

Percorsi di navigazione

L'organigramma dell'Ente può essere strutturato in un modo da consentire la selezione di diversi percorsi di navigazione. Questo dipende dal fatto che alcune tipologie di oggetti dell'organigramma oltre ad essere collegate tra di loro (e quindi navigabili secondo un preciso percorso del tipo parent-child) sono anche collegate direttamente all'Ente. Più precisamente, l'Ente ha legami diretti con AOO (attraverso il GruppoAOO), Struttura, Utente.

Sono individuabili tre possibili percorsi di navigazione (indipendentemente dalla molteplicità delle relazioni):

1. Ente, GruppoAOO, AOO, Struttura, Nodo, Utente
2. Ente, Struttura, Nodo
3. Ente, Utente

L'ordine con cui sono riportati gli oggetti è significativo del loro livello di annidamento.

Esempio: l'utente è collocato in un nodo che appartiene ad una struttura facente riferimento ad una AOO ...

Navigazione attraverso l'AOO

Gli identificatori impostabili come root sono del seguente tipo e l'ordine è significativo del livello di annidamento: null, Ente, GruppoAOO, AOO, Struttura, Nodo.

Utente non può comparire come root del presente servizio, indipendentemente dal percorso di navigazione, perché è posto all'ultimo livello di profondità.

Seguono alcuni esempi (ovviamente si presuppone che il principal, a seconda dei casi, abbia la visibilità adeguata alla navigazione richiesta):

- root = "sistema" (possono essere restituite tutte le tipologie di oggetti previste purché il principal sia amministratore di sistema: si veda, in proposito, il successivo paragrafo)
- root = identificatore Ente (possono essere restituiti GruppoAOO, AOO, Struttura, Nodo, Utente dello stesso Ente di collocazione del principal)
- root = identificatore GruppoAOO (possono essere restituiti AOO, Struttura, Nodo, Utente dello stesso Ente di collocazione del principal)

Il servizio può essere invocato in modo da iniziare la navigazione dal livello "sistema": cioè devono essere coinvolti tutti gli Enti configurati nella specifica installazione di Acta. In questo caso non è possibile valorizzare **rootNodeId** perché non esiste un identificatore utilizzabile allo scopo. Si consente quindi che il parametro rimanga non valorizzato: *null*. Il sistema però deve controllare che il principal abbia la visibilità a livello di sistema: solo l'amministratore di sistema può correttamente iniziare una navigazione a partire da tale livello.

La corretta valorizzazione del parametro *depth* consente di stabilire se navigare in discendenza oppure limitarsi al primo livello. In sostanza un *depth* == 1 equivale ad effettuare un'operazione simile alla *getChildren* prevista in altri moduli Acaris.

Dettaglio sulla valorizzazione del parametro *depth*:

- ✓ 1 solamente oggetti contenuti nel nodo identificato dall'identificatore root
- ✓ N in profondità per il numero di livelli specificato in 'N'
- ✓ -1 in profondità per tutti i livelli

Il sistema solleva apposita eccezione nel caso di valorizzazione non conforme del parametro in oggetto.

I valori **maxItem** e **SkipCount** permettono di ricevere i risultati di una navigazione a 'blocchi'.

Esempio la navigazione trova 30 oggetti ma il sistema chiamante decide che ne vuole ricevere 10 alla volta, il sistema chiamante non sa quanti oggetti ci sono in totale ma impostando **maxItem** = 10 e **SkipCount** = 0 alla prima chiamata riceverà i primi 10 oggetti ed il parametro **hasMoreItem** = 'true'.

Quindi potrà effettuare una seconda chiamata con **maxItem** = 10 e **SkipCount** = 10 ed il sistema ritornerà gli oggetti dal 11° al 20° e così via.

Seguono alcuni diagrammi di illustrazione di possibili percorsi di navigazione:

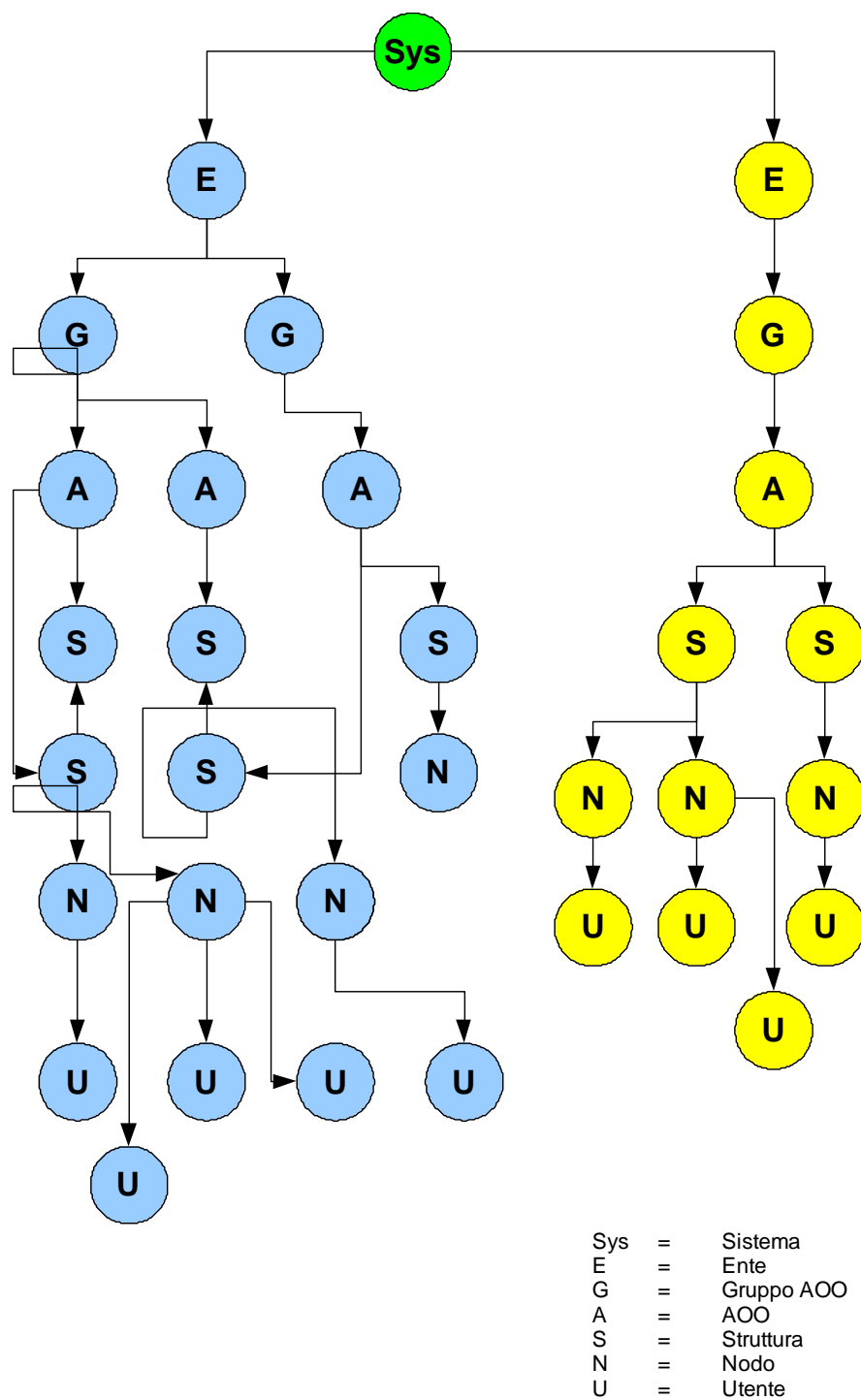


Figura 2 - tipologie oggetti getDescendants

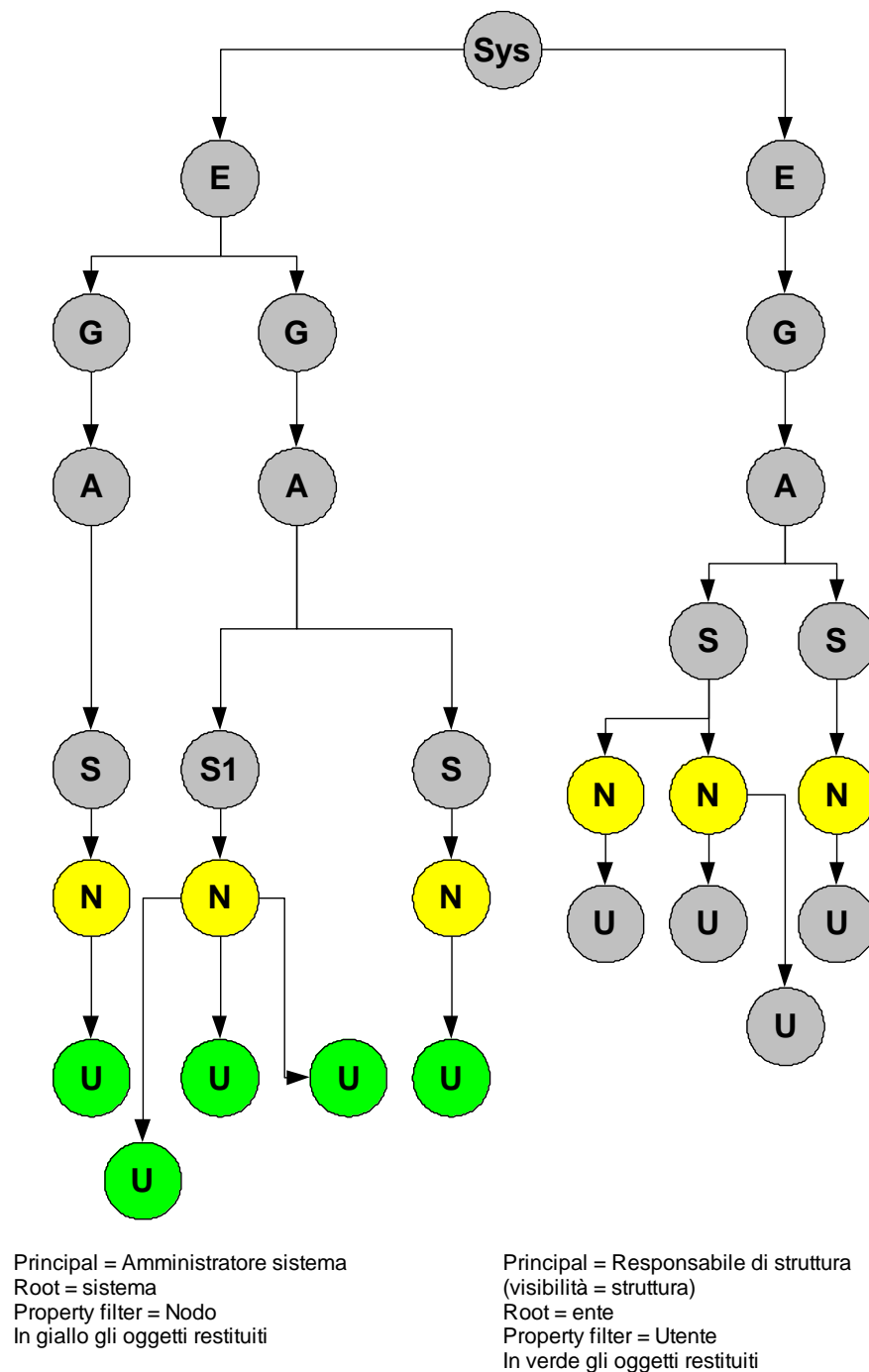
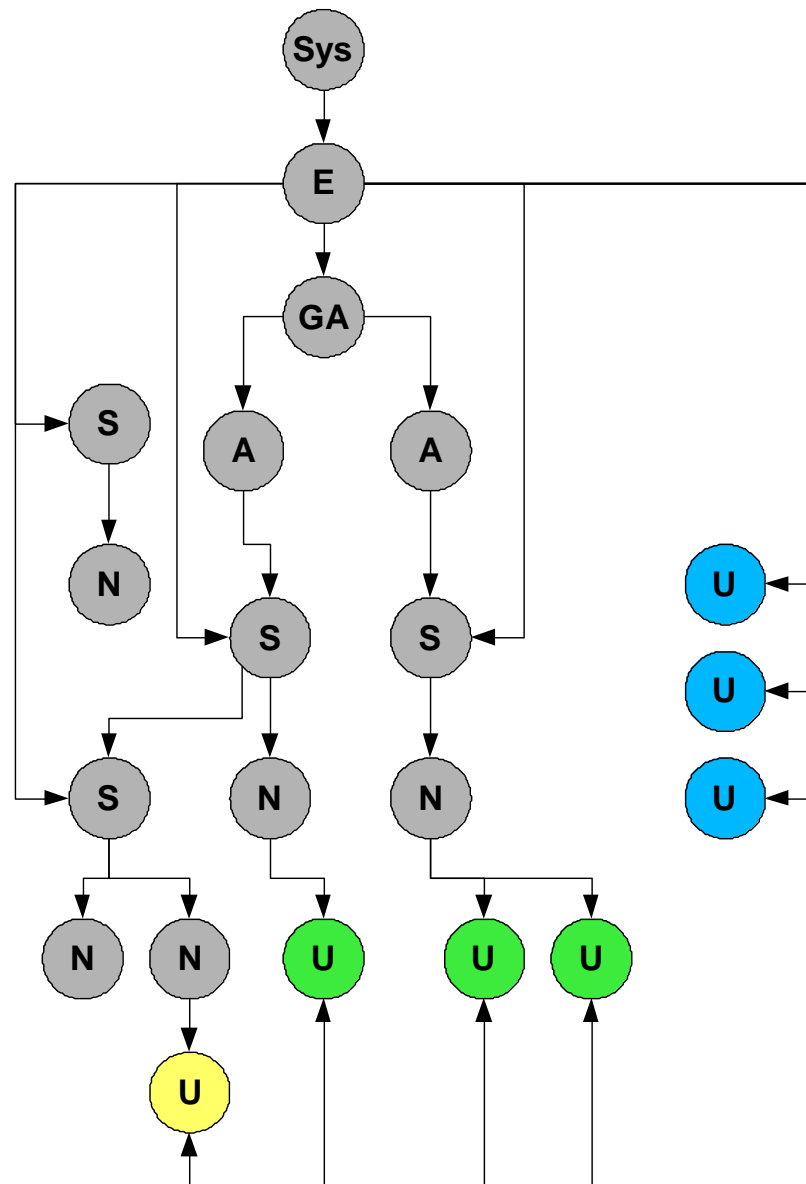


Figura 3 - getDescendants con esempi percorsi di navigazione



Principal = Responsabile di struttura (visibilità = struttura)

Root = ente

Property filter = Utente

In verde gli oggetti restituiti

In giallo gli oggetti non restituiti perché non collegati ad AOO e non raggiungibili tramite navigazione gerarchia

In blu gli oggetti non restituibili perché non collocati

Figura 4 - getDescendants con evidenziazione oggetti non restituibili

Navigazione dall'Ente alle Strutture

Valgono le regole già viste per il precedente scenario in quanto applicabili: valorizzazione parametri, paginazione, ...

A differenza di quanto stabilito nello scenario precedente, in questo caso, vengono solo navigati i percorsi che partono dall'Ente e passano per le strutture ma non per l'AOO. Con questo scenario vengono raggiunte le strutture collegate direttamente all'Ente ma che non hanno legami con AOO.

Il livello di annidamento è il seguente:

- “sistema”, Ente, Struttura, Nodo

In questo percorso di navigazione non vengono restituiti gli oggetti di tipo Utente perché non sarebbe assicurata la validità della collocazione dell'Utente attraverso il legame con Nodo-Struttura-AOO.

Seguono alcuni esempi relativi alla valorizzazione del parametro root:

- root = null (possono essere restituite tutte le tipologie di oggetti previste, secondo l'annidamento sopra riportato, purché il principal sia amministratore di sistema: si veda, in proposito, il successivo paragrafo)
- root = identificatore Ente (possono essere restituiti oggetti di tipo Struttura, Nodo dello stesso Ente di collocazione del principal)
- root = identificatore Struttura (possono essere restituiti oggetti di tipo Nodo dello stesso Ente di collocazione del principal)

Navigazione dall'Ente agli Utenti

Valgono le regole già viste per gli altri scenari in quanto applicabili.

In questo caso la navigazione parte dall'Ente ed è finalizzata alla restituzione degli utenti appartenenti allo stesso ma non collocati.

Il livello di annidamento è il seguente:

- “sistema”, Ente, Utente

Seguono alcuni esempi relativi alla valorizzazione del parametro root:

- root = null (possono essere restituiti oggetti del tipo Ente ed Utente)
- root = identificatore Ente (possono essere restituiti solo oggetti di tipo Utente dello stesso Ente di collocazione del principal)

Operazione getObjectParents

Descrizione dell'operazione

Operazione di interrogazione dell'organigramma per determinare i parents di un nodo. L'operazione di navigazione si interrompe all'Ente.

Interfaccia di richiamo (input)

- ✓ **ObjectIdType repositoryId:** Repository e quindi ente di riferimento.

- ✓ **PrincipalIdType principalId:** Utente che richiede il servizio.
- ✓ **ObjectIdType rootNodeId:** identificatore dell'unità organizzativa a partire dalla quale iniziare la navigazione.
- ✓ **date date:** serve per limitare la navigazione alle sole unità attive (valide) alla data passata come parametro.
- ✓ **PropertyFilterType filter:** permette di indicare tipologie di unità organizzative e proprietà da restituire

Interfaccia di output

- ✓ **<Array> ObjectResults:** Elenco di nodi dell'organigramma.
 - **ObjectIdType objectId:** Nodo dell'albero dell'organigramma.
 - **<Array> PropertyType properties:** Proprietà valorizzate per il nodo.

Il servizio permette di navigare in ascendenza l'organigramma di BKO a partire da un determinato nodo identificato come root. I risultati restituiti sono individuati a partire dal livello precedente a quello di root.

Gli identificatori impostabili come root sono del seguente tipo e l'ordine è significativo del livello di annidamento: Utente, Nodo, Struttura, AOO, GruppoAOO.

Se rootNodeid è uguale a Utente, il sistema tramite l'elenco delle sue identità e collocazioni, determina l'elenco dei Nodi organizzativi in cui è collocato e quindi inizia la navigazione in ascendenza recuperando gli oggetti parent incontrati ai vari livelli.

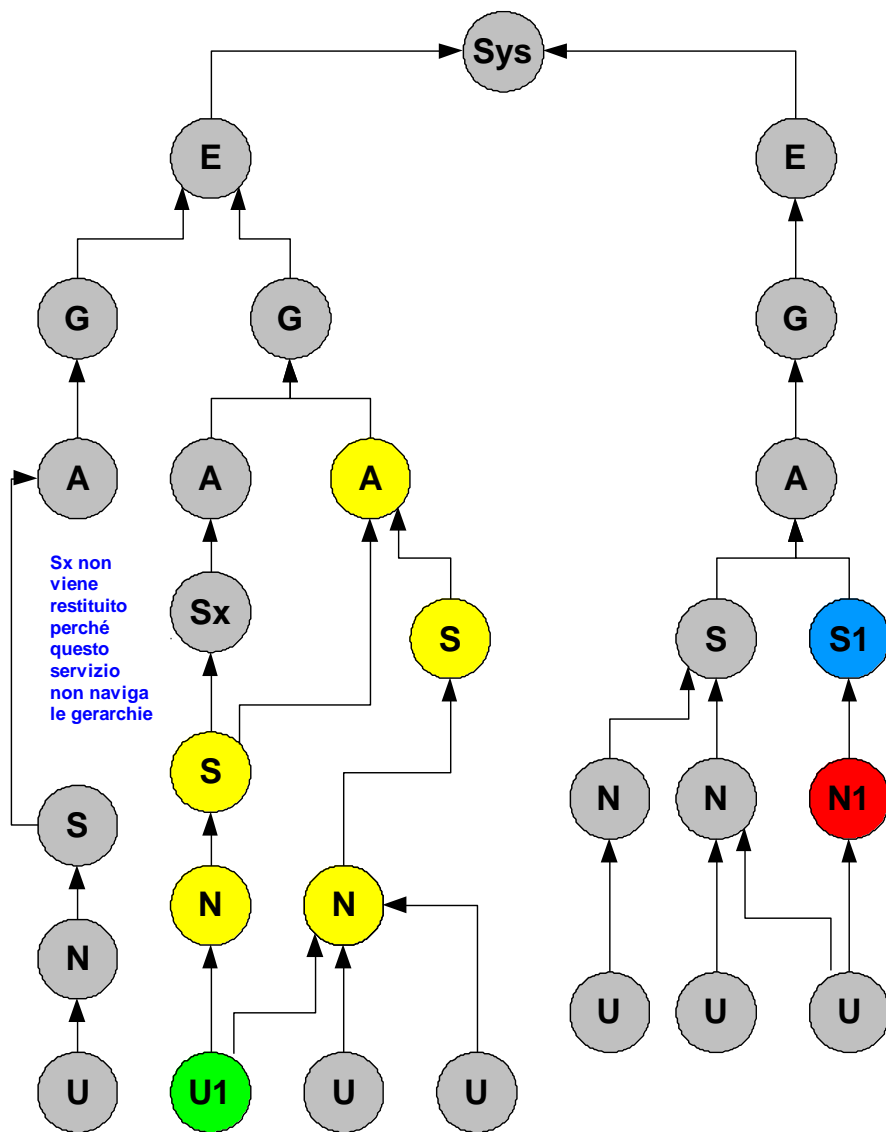
Anche in questo caso non vengono navigate le gerarchie tra strutture.

Seguono alcuni esempi (ovviamente si presuppone che il principal, a seconda dei casi, abbia la visibilità adeguata alla navigazione richiesta):

root == identificatore Utente (possono essere restituiti Nodo, Struttura, AOO, GruppoAOO, Ente)

Ente non può comparire come root del presente servizio perché è posto all'ultimo livello di ascendenza significativo per questa operazione.

Segue un diagramma illustrativo di possibili percorsi di navigazione in ascendenza.



Principal = Amministratore di ente (con
visibilità = ente)
Root = Utente 1 in verde
Property filter = Nodo Struttura e AOO
In giallo gli oggetti restituiti

Principal = Responsabile di struttura (con
visibilità = struttura)
Root = nodo 1 in rosso
Property filter = Struttura
In blu gli oggetti restituiti

Figura 5 - getObjectParents con esempio percorsi di navigazione

Operazione navigaGerarchia

Descrizione dell'operazione

Operazione di navigazione di una gerarchia tra strutture.

Questa operazione serve a navigare le gerarchie altrimenti non raggiungibili attraverso le operazioni di navigazione in ascendenza o discendenza di un organigramma.

Interfaccia di richiamo (input)

- ✓ **ObjectIdType repositoryId:** Repository e quindi ente di riferimento.
- ✓ **PrincipalIdType principalId:** Utente che richiede il servizio.
- ✓ **ObjectIdType objectId:** identificatore della struttura a partire dalla quale iniziare la navigazione della gerarchia
- ✓ **EnumHierarchyRole role:** indica il ruolo ricoperto dalla struttura nella relazione gerarchica che si intende navigare. I valori possibili sono i seguenti: PARENT, CHILD.
- ✓ **date date:** serve per limitare la navigazione alle sole gerarchie attive (valide) alla data passata come parametro.
- ✓ **PropertyFilterType filter:** permette di indicare le proprietà da restituire. L'entità di riferimento è sempre di tipo Struttura.

Interfaccia di output

- ✓ **<Array> ObjectIdType objectId:** identificatore oggetto.
- ✓ **<Array> PropertyType properties:** Proprietà valorizzate per l'oggetto restituito.

Logiche di Business

L'operazione in oggetto estrapola dalle operazioni di navigazione la specifica funzionalità di navigazione delle gerarchie costituite tra strutture.

Considerazioni sul parametro role

Consente di indicare in quale verso si vuole navigare la gerarchia. Segue il comportamento del sistema nei due casi presi in considerazione:

- **PARENT:** il sistema restituisce le strutture figlie relazionate a quella passata come parametro che viene considerata come struttura padre, indipendentemente dal livello gerarchico
- **CHILD:** il sistema restituisce l'ascendenza a partire dalla struttura passata come parametro che viene considerata come struttura figlia, fino ad arrivare alla struttura di primo livello in ascendenza

Esposizione delle informazioni relative alla struttura "parent" nelle gerarchie

Allo scopo di consentire la ricostruzione delle relazioni gerarchiche da parte dell'applicazione client, il sistema valorizza l'identificatore parent di ogni struttura restituita. Viene a questo scopo utilizzata la property "idParentStrutturaRelazionata" prevista nell'object model Acaris.

La property non sarà valorizzata per le strutture che non hanno un parent.

Operazione **getProperties**

Descrizione dell'operazione

Operazione di interrogazione puntuale dell'organigramma, questo servizio permette il recupero della collection properties di un oggetto specificato

- Ente
- Gruppo AOO
- AOO
- Struttura
- Nodo
- Utente
- Oggetti collegati ai precedenti e raggiungibili tramite gli appositi identificatori (relazioni in uscita)

Interfaccia di richiamo (input)

- ✓ **ObjectIdType repositoryId:** Repository e quindi ente di riferimento.
- ✓ **PrincipalIdType principalId:** Utente che richiede il servizio
- ✓ **ObjectIdType objectId:** Identificatore di cui restituire il dettaglio
- ✓ **PropertyFilterType filter:** Proprietà da valorizzare per il nodo interrogato.

Interfaccia di output

- ✓ **ObjectIdType objectId:** Nodo dell'albero dell'organigramma.
- ✓ **<Array> PropertyType properties:** Proprietà valorizzate per il nodo.

Logiche di Business

Operazione **getPropertiesMassive**

Descrizione dell'operazione

Nell'object model Acaris sono definiti alcuni oggetti con properties a cardinalità multipla. Si tratta essenzialmente delle relazioni con classi esterne con cardinalità 0..*. In questi casi, i servizi di navigazione e ricerca valorizzano properties di questo tipo con gli identificatori degli oggetti relazionati. Si rende necessario, quindi, un'ulteriore operazione di recupero del dettaglio di ogni singola entità considerata.

Allo scopo di facilitare le operazioni recupero ed ottimizzare le prestazioni complessive del sistema, in ogni modulo funzionale Acaris, viene messo a disposizione un servizio massivo di recupero delle properties.

Interfaccia di richiamo (input)

- ✓ **ObjectIdType repositoryId:** Repository e quindi ente di riferimento.
- ✓ **PrincipalIdType principalId:** Utente che richiede il servizio
- ✓ **ObjectIdType[] identificatori:** Elenco di identificatori di cui restituire il dettaglio
- ✓ **PropertyFilterType filter:** Proprietà da valorizzare per il nodo interrogato.

Interfaccia di output

- ✓ **<Array> ObjectResults:** Elenco di nodi dell'organigramma.
 - **ObjectIdType objectId:** Nodo dell'albero dell'organigramma
 - **<Array> PropertyType properties:** Proprietà valorizzate per il nodo.

Logiche di Business

Per quanto riguarda il parametro repositoryId non ci sono particolarità rispetto al funzionamento previsto in Acaris. Il parametro principalId viene utilizzato per verificare l'appartenenza dell'Utente all'Ente ricavabile da repositoryId: in caso di non corrispondenza il sistema deve sollevare apposita eccezione.

Il sistema restituisce il dettaglio delle entità rappresentate dai singoli identificatori.

Possono essere utilizzati solo identificatori di oggetti gestiti nel modulo BKO. Il sistema verifica l'appartenenza a tale modulo attraverso la relazione tra *acta_entita_acaris*, *acta_r_entita_acaris_modulo* e *acta_d_modulo_acaris*: se l'identificatore non si riferisce ad un'entità gestita nel modulo BKO il sistema solleva apposita eccezione.

Deve essere verificata anche la selezionabilità dell'entità corrispondente all'identificatore: a questo scopo viene fatta una verifica di come è stata valorizzata la colonna *flag_selezionabile* di *acta_c_entita_acaris*: se l'entità non è stata configurata come selezionabile il sistema solleva apposita eccezione.

Nel caso di problemi nel recupero del dettaglio delle entità richieste, il sistema adotta una politica conservativa:

- In caso di possibilità di restituire solo un sottoinsieme dei risultati, il sistema scarta gli altri e restituisce quelli disponibili: spetta al fruitore di capire quali entità non sono state restituite. Non viene restituita la causa del fallimento dell'operazione.
- Nel caso di impossibilità di restituire tutti i risultati per ragioni di merito (identificatore inesistente, ...) il sistema non restituisce risultati ma senza sollevare eccezione.

Per evitare di gestire il recupero di grosse moli di dati il servizio accetta in input un massimo di 50 identificatori.

Non è prevista la paginazione dei risultati perché il limite in oggetto è ampiamente gestibile per la costruzione dell'output. In caso di superamento del limite il sistema solleva apposita eccezione.

Operazione **getPrincipalExt**

Descrizione dell'operazione

Questa operazione consente di recuperare l'elenco degli eventuali "principal" che possono operare sul repository di interesse per conto di una determinata applicazione client.

Interfaccia di richiamo (input)

- ObjectIdType: repositoryId
- CodiceFiscaleType: codiceFiscale
- IdAooType: idAoo
- IdStrutturaType: idStruttura
- IdNodoType: idNodo
- ClientApplicationInfo: clientApplicationInfo

Interfaccia di output

- Elenco di oggetti di tipo PrincipalExtResponseType

Eccezioni specifiche

SERGEN-E004=ACARIS - Fruitore non presente nel sistema

SERGEN-E005=ACARIS - AppKey non autorizzata all'utilizzo dei servizi ACARIS

SERGEN-E006=ACARIS - Fruitore {0} non abilitato all'utilizzo del servizio {1}

Logica di Business

Si richiama per quanto compatibile la logica di getPrincipal.

Lo scopo del servizio è quello di restituire un, eventuale, elenco di “principal”: si tratta sostanzialmente della collocazione di un utente in un determinato nodo organizzativo dell'ente e dell'elenco dei profili con i quali opera nella stessa (es. archivista).

Per agevolare il fruitore, in questa versione del servizio, per ogni tripletta aoo-struttura-nodo restituita dal sistema, vengono valorizzati identificatore, codice e descrizione: il fruitore così ha tutti gli elementi per discriminare nel caso di restituzione di una collection di risultati.

La differenza più importante rispetto al servizio getPrincipal e rappresentata dall'obbligo per il fruitore di valorizzare un parametro di tipo ClientApplicationInfo.

Ogni fruitore che utilizza i servizi ACARIS deve avere un'apposita application key (fornita dal backoffice ACTA) che viene utilizzata, insieme agli altri parametri, per il recupero dell'elenco dei principal appartenenti all'utente. Il sistema verifica l'autorizzazione dell'application client all'utilizzo dei servizi ACARIS e, in caso di verifica positiva, per ogni principal restituito veicola, in modo trasparente al fruitore, anche la chiave dell'application client.

Ad ogni successiva invocazione di servizio, tramite il principal speso dal fruitore, il sistema potrà identificare l'application client ed effettuare le verifiche previste.

La property “info” di tipo VarargsType contenuta nel parametro “clientApplicationInfo” (di tipo ClientApplicationInfo) ha lo scopo di fornire al sistema informazioni aggiuntive sulla client application e deve essere valorizzata con le chiavi e i valori forniti dal back office di ACTA.

Per evitare interventi sul codice dell'applicazione client in caso di modifiche riguardanti i valori veicolati dalla property in oggetto, si consiglia di recuperare i valori da un repository locale all'applicazione e di istanziare e valorizzare automaticamente il parametro: la gestione quindi potrà essere trasparente per il fruitore.

Nell'attuale versione non sono presenti chiavi.

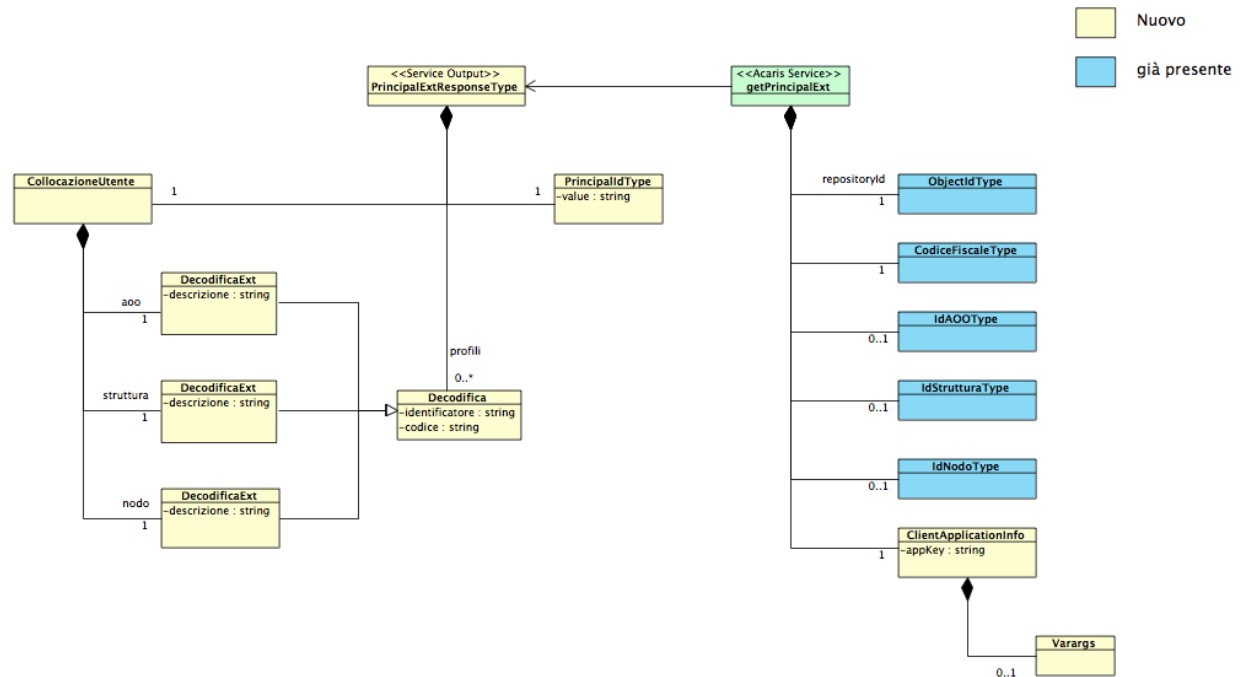


Figura 6 – interfaccia servizio getPrincipalExt

Governance

Requisiti di sicurezza

I requisiti di qualità sono comuni all'intero sistema e sono elencati nel documento di requisiti funzionali (cfr. riferimento [A1]).