

**ACTA**  
*Servizi ACARIS*

**Manuale Integrazione Fruitore**

**Versione 2.0**

## STATO DELLE VARIAZIONI

<b>VERSIONE</b>	<b>DATA VALIDITÀ</b>	<b>PARAGRAFO O PAGINA</b>	<b>DESCRIZIONE DELLA VARIAZIONE</b>
02	08/06/2020	Cap.3,4,5	Inseriti capitoli relative alle specifiche tecniche
01	30/04/2020	Intero documento	Versione iniziale

## Sommario

Sommario	2
1 Introduzione	5
1.1 Scopo del documento	5
1.2 Prerequisito	5
1.3 Riferimenti	5
2 Principali scenari d'uso dei servizi	7
2.1 Creazione di una Registrazione di protocollo “in arrivo” ed archiviazione in una specificata struttura aggregativa	7
2.2 Ricerca classificazioni allegate di una classificazione principale	7
2.3 Inserimento di un documento in una precisa struttura aggregativa	7
2.4 Inserimento di un documento “allegato”	7
2.5 Recupero del content-stream di un Document a partire da una classificazione	8
2.6 Creazione di una Registrazione di protocollo “in partenza”	8
2.7 Ricerca protocolli per intervallo date	8
3 Specifiche tecniche di integrazione	9
3.1 Object Model	9
3.1.1 Modelli per l'utilizzo di “properties”	9
3.2 Definizione interfaccia	10
3.2.1 Approccio “Contract First”	10
3.2.2 Messaggi	10
4 Riferimenti Tecnologici	11
4.1 Specifiche	11
4.2 Framework	11
4.3 Librerie	11
4.4 Libreria di implementazione dell'Object Model	11
4.5 Interfaccia Wrapper Web Services	11
4.6 AcarisException	11
4.7 Eccezioni e gestione Fault in ACARIS	12
4.8 Gestione allegati	12
4.9 DTO	12
5 Il catalogo dei servizi e delle operazioni	13
5.1 Modalità di esposizione dei servizi ACARIS	13
5.2 Fruizione internet APIStore	13
6 RepositoryService	14
6.1 L'operazione getRepositories	14
7 BackOfficeService	15
7.1 L'operazione getPrincipalExt	15

8	ObjectService	16
8.1	L'operazione query	16
8.2	L'operazione updateProperties	17
8.3	L'operazione moveDocument	17
9	DocumentService	19
9.1	L'operazione trasformaDocumentoPlaceholderInDocumentoElettronico	19
9.2	Oggetti comuni	20
10	NavigationService	22
10.1	L'operazione getChildren	22
11	RelationshipService	22
11.1	L'operazione getObjectRelationship	22
12	OfficialBookService	23
12.1	L'operazione creaRegistrazione	23
13	MultifilingServices	26
13.1	L'operazione addAssociativeObjectToFolder	26
13.2	L'operazione aggiungiClassificazione	27
14	ManagementService	30
14.1	L'operazione getVitalRecordCode	30
14.2	L'operazione query	30
15	SubjectRegistryService	30
15.1	L'operazione creaSoggetto	30
15.1.1	Output	31
15.2	L'operazione getProperties	31
15.2.1	Output	31
15.3	L'operazione updateProperties	32
15.3.1	Output	32
15.4	L'operazione query	32
15.4.1	Output	33
15.5	L'operazione ricercaSoggetto	33
15.6	L'operazione ricercaSoggettoDaFonteEsterna	34
15.6.1	Output	34
15.7	L'operazione getPropertiesMassive	34
15.7.1	Output	35
16	SMSService	35
16.1	L'operazione query	35
16.2	L'operazione creaSmistamento	35
16.2.1	Output	36
17	Oggetti comuni a tutti i servizi ACARIS	36

## Indice delle tabelle

Tabella 1: Dati output getRepositories.....	14
Tabella 2: Dati input getPrincipalExt.....	15
Tabella 3: Dati output getPrincipalExt.....	15
Tabella 4: Dati input query .....	16
Tabella 5: Dati output query .....	17
Tabella 6: Dati input updateProperties.....	17
Tabella 7: Dati output updateProperties.....	17
Tabella 8: Dati input moveDocument.....	17
Tabella 9: Dati output moveDocument.....	18
Tabella 10: Dati input trasformaDocumentoPlaceholderInDocumentoElettronico.....	19
Tabella 11: Dati output trasformaDocumentoPlaceholderInDocumentoElettronico.....	19
Tabella 12: DocumentoFisicoIRC .....	20
Tabella 13: ContenutoFisicoIRC .....	20
Tabella 14: FailedStepInfo.....	21
Tabella 15: RelationshipPropertiesType.....	21
Tabella 16: Dati Input getChildren .....	22
Tabella 17: Dati output getChildren.....	22
Tabella 18: Dati Input creaRegistrazione.....	23
Tabella 19: Dati di output creaRegistrazione.....	24
Tabella 20: Protocollazione .....	24
Tabella 21: ProtocollazioneDocumentoEsistente .....	24
Tabella 22: ProtocollazioneDaSmistamento .....	24
Tabella 23: RegistrazioneAPI .....	24
Tabella 24: RegistrazioneArrivo.....	25
Tabella 25: DocumentoFisicoPropertiesType.....	37
Tabella 26: ContenutoFisicoPropertiesType.....	37
Tabella 27: AcarisContentStreamType.....	37
Tabella 28 QueryConditionType .....	37
Tabella 29 NavigationConditionInfoType .....	37
Tabella 30: PropertyFilterType.....	37
Tabella 31: PagingReponseType .....	38
Tabella 32: QueryNameType.....	38
Tabella 33: ObjectResponseType .....	38
Tabella 34: PropertyType .....	38
Tabella 35: SimpleResponseType.....	38

## 1 Introduzione

I servizi “ACTA Archive Interoperability Services” (in seguito ACARIS) permettono a tutti gli applicativi verticali che producono e/o gestiscono documenti di effettuare l’integrazione con ACTA, sistema di gestione documentale di archivio e protocollo del CSI Piemonte.

### 1.1 Scopo del documento

Il documento è indirizzato ai fruitori dei servizi applicativi del sistema documentale ACTA. Lo scopo è quello di fornire indicazioni per una integrazione rapida tra un sistema alimentante e il sistema documentale e di essere d’aiuto in fase di stima di progetto.

### 1.2 Prerequisito

Prerequisito essenziale e obbligatorio per l’integrazione con ACARIS è aver contattato il responsabile referente del servizio dell’ente di riferimento.

Dovrà quindi essere redatta una analisi archivistica in cui si delinea l’esigenza dell’applicativo integrante e si definisce la logica di integrazione.

Per avviare la fase di sviluppo dell’integrazione, è necessario contattare il supporto tecnico ACARIS tramite invio di un messaggio di posta all’indirizzo e-mail [supporto\\_acaris@csi.it](mailto:supporto_acaris@csi.it).

### 1.3 Riferimenti

Si riportano di seguito i documenti relativi alle specifiche di dettaglio, disponibili in aggiunta al presente manuale ottenibili contattando [supporto\\_acaris@csi.it](mailto:supporto_acaris@csi.it).

Numero	Nome file	Descrizione
[1]	PGED-ACTA-STE011-AcArIs-V04-BackOfficeServices	Descrizione dei servizi esposti da BackOffice service
[2]	PGED-ACTA-STE011-AcArIs-V09 ObjectServices	Descrizione dei servizi esposti da Object service
[3]	PGED-ACTA-STE016-AcArIs-V10-DocumentServices	Descrizione dei servizi esposti da Document service
[4]	PGED-ACTA-STE016-AcArIs-V02-RelationshipServices	Descrizione dei servizi esposti da Relationship service
[5]	PGED-ACTA-STE015-AcArIs-V06-NavigationServices	Descrizione dei servizi esposti da Navigation service
[6]	ACARISWS-BackOfficeService.wsdl, ACARISWS-DocumentService.wsdl, ACARISWS-ManagementService.wsdl, ACARISWS-MultifilingService.wsdl, ACARISWS-NavigationService.wsdl, ACARISWS-ObjectService.wsdl, ACARISWS-OfficialBookService.wsdl, ACARISWS-RelationshipsService.wsdl, ACARISWS-RepositoryService.wsdl, ACARISWS-SMSService.wsdl, ACARISWS-SubjectRegistryService.wsdl ACARIS-ArchiveMessaging.xsd, ACARIS-BackOfficeMessaging.xsd, ACARIS-DocumentMessaging.xsd, ACARIS-ManagementMessaging.xsd, ACARIS-OfficialBookMessaging.xsd, ACARIS-SMSMessaging.xsd, ACARIS-SubjectRegistryMessaging.xsd	Operazioni dei servizi ACARIS

[7]	ACARIS-Common.xsd ACARIS-Common-PRT.xsd ACARIS-Archive.xsd ACARIS-BackOffice.xsd ACARIS-Document.xsd ACARIS-Management.xsd ACARIS-OfficialBook.xsd ACARIS-SMS.xsd ACARIS-SubjectRegistry.xsd	Descrizione dei tipi di dati utilizzati
[8]	Apache CXF 2.2.3	
[9]	ACARIS-binding-archive.xjb ACARIS-binding-backoffice.xjb ACARIS-binding-document.xjb ACARIS-binding-management.xjb ACARIS-binding-officalbook.xjb ACARIS-binding-sms.xjb ACARIS-binding-subjectregistry.xjb	

## 2 Principali scenari d'uso dei servizi

Di seguito sono illustrati, in modo schematico, alcuni degli scenari tipici di interazione di una applicazione client attraverso i servizi ACARIS. Questa non vuole essere una illustrazione esaustiva delle modalità in cui possono essere utilizzati ed orchestrati le operazioni esposte ma offrire una indicazione sul flusso logico degli scenari di utilizzo più comuni.

### 2.1 Creazione di una Registrazione di protocollo “in arrivo” ed archiviazione in una specifica struttura aggregativa

1. Invocazione del servizio *creaRegistrazione* (vd 12.1) indicando la volontà di creare un oggetto di tipo “in arrivo” senza l’indicazione di un documento esistente in archivio.
2. Il sistema, in base ai parametri con cui è invocato il servizio *creaRegistrazione*, crea una registrazione di protocollo, una classificazione principale ed eventualmente degli allegati “place holder” (a cui non è collegato un contenuto).
3. Aggiornare i metadati della classificazione principale creata utilizzando l’operazione *updateProperties* (vd 8.2) di Object service.
4. Se si vuole associare alla classificazione principale un contenuto, invocazione del servizio *trasformaDocumentoPlaceholderInDocumentoElettronico* (vd 9.1) utilizzando come parametro di ingresso l’identificativo della classificazione principale ed il contenuto da associare.
5. Se sono stati create delle classificazioni allegate, ricerca degli identificativi delle classificazioni allegate (vd 2.2) e, per ognuno di questi, effettuare i passi 3 e 4.
6. Per ottenere l’identificativo della struttura aggregativa in cui archiviare il documento, effettuare la ricerca tramite il servizio *query* (vd 8.1) di Object service, per oggetti dello specifico tipo di Folder (per esempio, Serie tipologica di documenti, Fascicolo Reale Annuale)
7. Con le informazioni del Folder (aggregazione) target, utilizzo dell’identificativo univoco (Object ID) del Folder come parametro di ingresso del servizio *moveDocument* (vd 8.3) di Object service, in modo da spostare la classificazione principale nell’aggregazione ricercata.

### 2.2 Ricerca classificazioni allegate di una classificazione principale

1. Invocazione del servizio *getChildren* (vd 10.1) di Navigation service utilizzando come parametro di ingresso l’identificativo della classificazione principale: si otterrà l’identificativo di un oggetto di tipo Gruppo Allegati.
2. Invocazione del servizio *getChildren* utilizzando come parametro di ingresso l’identificativo del gruppo allegati per ottenere la lista delle classificazioni allegate.

### 2.3 Inserimento di un documento in una precisa struttura aggregativa

1. Ricerca tramite il servizio *query* di Object Services, per oggetti dello specifico tipo di Folder (per esempio, Serie tipologica di documenti, Fascicolo Reale Annuale)
  - a. In caso di esito negativo, invocazione del servizio *createFolder* di Object Services per l’inserimento nel repository di un Folder del tipo e ottenimento delle sue informazioni.
2. Con le informazioni del Folder target, utilizzo dell’identificativo univoco (Object ID) del Folder come parametro di ingresso del servizio *creaDocumento* di Document Services.

Il servizio *creaDocumento*, in modo “atomico”, orchestra lo stesso flusso logico delle invocazioni di *createAssociativeDocument* e *createRelationships* di Object Service: si crea così un oggetto Folder (della sottoclasse Classificazione) ed i necessari oggetti sottoclasse di Document (Documento archivistico, Contenuto Fisico) per caricare un content stream. Il servizio mette in relazione la classificazione con l’oggetto derivato da Folder (la struttura aggregativa) ricercato.

### 2.4 Inserimento di un documento “allegato”

1. Ricerca tramite il servizio *query* di Object Services del Folder a cui è collegato il documento

- principale a cui si vuole allegare un documento, la ricerca è effettuata per specifico tipo di Folder (per esempio Serie tipologica di documenti, Fascicolo Reale Annuale).
2. Ricerca tramite il servizio query di Object Service, per oggetto della specifica sottoclasse di Folder (per esempio, Documento Semplice) che rappresenta il documento principale, al fine di ottenere il suo identificativo.
    - a. Per limitare la ricerca del documento, indicare come parametro di ricerca l'identificativo del Folder (struttura aggregativa) con cui il documento principale è in relazione padre-figlio
  3. Utilizzando l'identificativo (Object ID) del documento semplice risalire all'oggetto classificazione a cui è collegato tramite il servizio getObjectRelationships di Navigation Services. La classificazione è un oggetto derivato da Folder che indica la relazione tra un documento archivistico (sottoclasse di Document) ed una struttura aggregativa.
  4. A partire dall'oggetto classificazione utilizzare il suo identificativo come parametro di ingresso del servizio creaDocumento di Document Services.

Il servizio creaDocumento, in questo scenario, crea ed inserisce un oggetto Folder (della sottoclasse Classificazione) ed i necessari oggetti sottoclasse di Document (Documento archivistico, Contenuto Fisico) e crea una relazione con la classificazione principale. Nel caso specifico, l'oggetto di tipo Folder, che modella la relazione padre-figlio tra una classificazione principale ed una classificazione allegata, appartiene alla sottoclasse Gruppo Allegati ed è specifico per modellare la relazione.

## **2.5 Recupero del content-stream di un Document a partire da una classificazione**

1. A partire dall'identificativo di un oggetto Folder (sottoclasse Classificazione), invocazione del servizio getChilden di Navigation Services per ottenere l'identificativo del Document (sottoclasse Documento archivistico)
2. Invocazione del servizio getObjectRelationships di Relationships Services per ottenere l'identificativo del Folder (sottoclasse Documento Fisico)
3. Per ottenere il content stream ricercato, utilizzare il servizio getContentStream di Object Service utilizzando come parametro di ingresso l'identificativo di tipo Documento Fisico.

## **2.6 Creazione di una Registrazione di protocollo “in partenza”**

1. Nel caso in cui il documento a cui si vuole associare una registrazione di protocollo non sia presente nel repository, effettuare l'inserimento del documento nella struttura aggregativa desiderata (vd 2.3.1)
  - a. Eventualmente, inserimento dei suoi documenti allegati (vd 2.3.2)
2. Invocazione del servizio creaRegistrazione di Official Book Services indicando la volontà di creare un oggetto di tipo “in partenza” a partire da documenti già esistenti in repository (utilizzando la modalità “Protocollazione da documenti esistenti”) e passando, tra i parametri di ingresso, l'identificativo univoco della classificazione principale (ottenuto come risultato dell'invocazione del servizio creaDocumento o tramite ricerca).

## **2.7 Ricerca protocolli per intervallo date**

Per leggere le registrazioni di protocollo presenti nel sistema ed i dati collegati è necessario orchestrare diversi servizi.

1. Nel caso della ricerca per date di protocollo si utilizza il servizio officialbookService.query().
2. Successivamente per risalire ai dati delle classificazioni legate ad una registrazione di protocollo e navigare le relazioni tra classificazione principale ed eventuali allegati, si devono orchestrare invocazioni dei servizi officialbookService.query(), objectService.query(), navigationService.getChildren(), navigationService.getObjectRelationships().
3. Per scaricare il file collegato ad un documento, utilizzare objectService.getContentStream().

### 3 Specifiche tecniche di integrazione

Le funzioni di business di Acta sono fruibili da applicazioni esterne tramite l'invocazione di servizi applicativi.

I servizi applicativi di Acta sono stateless e ogni chiamata agli stessi rappresenta un'unità atomica di lavoro. Nel caso in cui sia necessario mantenere uno stato fra una chiamata e l'altra, tale operazione è a carico dell'applicazione fruitrice.

I servizi vengono definiti avendo come riferimento lo standard CMIS.

I riferimenti sono il data model e le interfacce previste dallo standard: entrambi sono stati personalizzati allo scopo di adattarli alle esigenze specifiche di ACTA.

Il data model ha ispirato il corrispondente "object model" dei servizi applicativi ACTA.

Le interfacce, opportunamente adattate, costituiscono la base per la definizione dell'interfaccia di esposizione dei servizi. I servizi del sottomodulo di "Management", però, essendo specifici del business di ACTA non trovano riscontro nello standard CMIS.

#### 3.1 Object Model

Definisce la rappresentazione di tutti gli oggetti (ed oggetti da questi referenziati) che compaiono nelle firme dei servizi esposti tramite l'interfaccia pubblica degli stessi.

L'Object Model viene definito attraverso schemi espressi in W3C XML Schema syntax.

La previsione di esposizione dei servizi applicativi, anche, come Web Services, condiziona in tal senso e guida la definizione dell'object model e della sua struttura.

Tale scelta, obbligata a causa del riferimento allo standard CMIS, consente di riferire il modello all'utilizzo di standard riconosciuti con il vantaggio di poter utilizzare tools che consentono, a partire dal modello, di realizzare in automatico il binding con il linguaggio scelto per l'implementazione: le classi Java di implementazione del modello vengono generate a partire dalle definizioni contenute negli schemi XML. Analoga procedura viene seguita per le interfacce dei servizi che vengono generate a partire da appositi WSDL.

Opportune annotazioni consentono di utilizzare compilatori per eseguire adattamenti durante il processo di binding; questo avviene in modo trasparente: cioè essi non sono resi pubblici attraverso il contratto (XSD e WSDL) rilasciato ai fruitori.

##### 3.1.1 Modelli per l'utilizzo di "properties"

Alcune operazioni previste dallo standard CMIS (ad esempio quelle di creazione di oggetti) prevedono un parametro denominato properties.

Lo standard prevede che il parametro possa essere modellato in modo funzionale alle esigenze dello specifico dominio di applicazione.

La modellazione può basarsi su diverse soluzioni delle quali, però, le più probabili sono:

- Applicazione di un modello generico
- Applicazione di un modello tipizzato

Per modello generico s'intende la previsione di un parametro nel quale sono appiattiti i riferimenti al tipo di dato, al valore ed alla classe di appartenenza.

Per modello **tipizzato** s'intende la previsione dell'utilizzo di strutture dati specifiche per la rappresentazione dei corrispondenti oggetti di dominio: es. folder, voce, aggregazione, fascicolo...

L'object model ACARIS è basato su un modello tipizzato.

### 3.2 Definizione interfaccia

Nel disegno delle interfacce esposte da servizi possono essere adottati due modelli di sviluppo che in letteratura sono così definiti:

- “**Start from Java**” oppure “Bottom Up” nel caso di esposizione come servizi di logiche implementate in classi già esistenti
- “**Start from WSDL**” oppure “Top Down” oppure “Contract First” nel caso di esposizione come servizi di interfacce generate tramite un processo che utilizza WSDL preesistenti

#### 3.2.1 Approccio “Contract First”

Il riferimento alla specifica CMIS ha reso obbligatoria la scelta di tale modello di sviluppo.

La soluzione, coerentemente con l’approccio CMIS di definizione delle interfacce, privilegia il contratto con i fruitori esposto tramite Web Services

Le interfacce CMIS sono già definite nella specifica: ACARIS le ha adattate al contesto ACTA ed ha modellato tramite XML Schema l’*Object Model* che rappresenta l’estensione del *data model* CMIS ai fini della rappresentazione del particolare “*domino*” di applicazione.

Il contratto (interfacce esposte ai fruitori dei servizi, soprattutto Web Services) è costituito tramite WSDL, scaricabile da apposito url secondo quanto stabilito dalle specifiche in tema di esposizione Web Services. Sono previsti i WSDL di definizione dei seguenti servizi:

Management, Multifiling, Navigation, Object, Relationships, Repository, Document, OfficialBook, BackOffice, SMS, SubjectRegistry.

Attraverso il WSDL del singolo servizio il fruitore, effettuando un binding con il proprio linguaggio di implementazione, genera una libreria di classi locale che utilizza per invocare le operazioni esposte dal singolo servizio. Per i fruitori che implementano l’integrazione utilizzando il linguaggio Java è resa comunque disponibile una libreria (actasrv-client.jar) che contiene le classi generate nella modalità appena descritta a partire dal WSDL utilizzando il tool **wsdl2java**.

Per non appesantire il contratto con dettagli riguardanti il *binding* con un particolare linguaggio, ACARIS segue l’approccio *external binding declaration*.

Il *SOAP message* in input e output contiene nel body di *envelope* la rappresentazione XML (*marshalling*) del tipo definito nell’XML schema. Il framework sul quale viene basata l’implementazione del servizio effettua un’operazione di *unmarshalling* del messaggio SOAP e costruisce delle classi Java contenute nella libreria costituente l’object model ACARIS.

Per la documentazione relativa fare riferimento al paragrafo 1.3 e al paragrafo 4 **Riferimenti Tecnologici**

#### 3.2.2 Messaggi

Per lo schema di definizione fare riferimento ai documenti al paragrafo 1.3.

Questo schema elenca:

- *Complex types* costituenti le sezioni “Request” e “Response” di ogni messaggio
- Definizioni di tipi che compaiono nelle segnature dei messaggi ma non fanno direttamente parte dell’object model
- La definizione di AcarisException e AcarisFaultType

I diversi WSDL utilizzano i messaggi per la costruzione delle operazioni che costituiscono l’interfaccia di ogni servizio.

## 4 Riferimenti Tecnologici

### 4.1 Specifiche

- JAX-WS 2.x  
si vedano i riferimenti di letteratura per quanto riguarda le specifiche SUN in tema di realizzazione di Web Services e la documentazione specifica di [8]
- JAXB 2.x  
rappresenta la specifica riferita come standard da JAX-WS per effettuare il *binding* Java/XML in modo da rappresentare *Schema components* (elementi, tipi, attributi) come *Java contents* (classi, bean properties)
- C.S.I.

### 4.2 Framework

CXF 2.2.3: rappresenta il framework adottato da CSI per l'introduzione delle specifiche JAX-WS 2.x nell'attuale stack tecnologico per l'implementazione della SOA.

### 4.3 Librerie

Per quanto riguarda il Wrapper WS (Web Services) si faccia riferimento alle librerie utilizzate in [8].

Poiché il processo di generazione automatica dell'object model, a partire dai WSDL dei servizi, è attività che prescinde dalla modalità di esposizione, in questo allegato viene fatto un espresso riferimento alla libreria **jaxb-api-2.1.jar**

### 4.4 Libreria di implementazione dell'Object Model

Il tool **wsdl2java** previsto in CXF [8] può effettuare la generazione sia dei value object corrispondenti ai tipi definiti nei WSDL/XSD passati in input (object model), che delle interfacce esportate dai singoli servizi sulla base delle definizioni contenute nei WSDL: *Service Endpoint Interface*.

Attraverso apposite istruzioni contenute in [9] è possibile personalizzare l'alberatura di package delle classi generate in modo da renderla compatibile con le linee guida CSI in merito.

La libreria di classi così generata deve essere disponibile all'implementazione dei servizi ACARIS e disponibile nel relativo classpath.

### 4.5 Interfaccia Wrapper Web Services

Utilizza la *Service Endpoint Interface* (SEI) di ogni servizio così come generata dall'apposito tool a partire dal WSDL di ogni servizio.

La SEI viene implementata dal relativo *Service Implementation Bean* (SIB) il cui compito, esauriti eventuali compiti di adattamento, è quello di invocare il vero e proprio *ServiceImpl* per l'esecuzione delle logiche di servizio.

### 4.6 AcarisException

Viene definita in [6] insieme alle definizioni dei "messaggi" costituenti rispettivamente la parte "*Request*" e "*Response*" di ogni operazione.

AcarisException viene definita come un element di tipo *AcarisFaultType*:

```
<xs:complexType name="acarisFaultType">
  <xs:sequence>
    <xs:element name="exceptionType" type="tns:enumServiceException" />
    <xs:element name="errorCode" type="tns:enumErrorCodeType" />
    <xs:element name="technicalInfo" type="tns:string" />
    <xs:element name="objectId" type="tns:ObjectIdType" />
    <xs:element name="className" type="tns:enumObjectType" />
    <xs:element name="propertyName" type="tns:string" />
  </xs:sequence>
</xs:complexType>
```

```
</xs:sequence>  
</xs:complexType>  
<xs:element name="acarisException" type="tns:acarisFaultType" />
```

#### 4.7 Eccezioni e gestione Fault in ACARIS

AcarisException viene utilizzata in funzione di “*fault*” nei WSDL di definizione dei singoli servizi. Si rimanda alla letteratura su SOAP sulla definizione del contenuto di *envelope* per un approfondimento sullo scopo ed il funzionamento di *SOAP fault message*.

In ACARIS, il tool *wSDL2java*, attraverso le definizioni contenute nei singoli WSDL dei servizi, genera in automatico la segnatura della singola operazione utilizzando quanto definito come “*fault*” in funzione di eccezione sollevata dalla stessa operazione.

Vanno inserite nell'alberatura standard: AcarisException, nella versione generata in automatico da *wSDL2java*, quindi, viene inserita in `it.doqui.acta.actasrv.exception`. Questa versione viene utilizzata dal ServiceImpl e dal wrapper SOAP.

#### 4.8 Gestione allegati

In ACARIS la gestione degli allegati è effettuata tramite la classe *AcarisContentStreamType* definita come segue in [6].

```
<!-- Content Stream -->  
<xs:complexType name="acarisContentStreamType">  
  <xs:sequence>  
    <xs:element name="length" type="tns:contentStreamLengthType" />  
    <xs:element name="mimeType" type="tns:enumMimeTypeType" minOccurs="0" />  
    <xs:element name="filename" type="tns:contentStreamFilenameType" minOccurs="0" />  
    <xs:element name="streamMTOM" type="tns:base64Binary"  
      xmime:expectedContentTypes="*/*" />  
    <xs:element name="stream" type="tns:base64Binary" />  
  </xs:sequence>  
</xs:complexType>
```

La classe è modellata in modo da gestire il passaggio dei parametri compatibilmente con la doppia modalità di esposizione.

#### 4.9 DTO

Le classi che rappresentano l'*object model* di ACARIS devono essere assegnate al layer apposito all'interno della componente utilizzata.

Esempio: `it.doqui.acta.actasrv.dto.acaris.common`

In sede di generazione automatica delle classi Java a partire dai *wSDL*, tramite una *binding declaration* in [9], è possibile decidere conformemente l'alberatura di package dei DTO costituenti l'*object model*.

I DTO, specifici di una particolare modalità di esposizione, possono essere inseriti in un package **dto** apposito dentro il wrapper di riferimento: ad esempio i DTO usati per il mapping tra modello tipizzato e generico usati nel “*generic wrapper*”.

Esempio: `it.doqui.acta.actasrv.util.acaris.wrapper.dto.common.PropertyType`

## 5 Il catalogo dei servizi e delle operazioni

I servizi sono fruibili sia da esposizione su rete Internet che su rete interna CSI.

Le operazioni messe a disposizione sono suddivise in diversi servizi che le raggruppano logicamente e che sono riconducibili ai moduli di ACTA.

- RepositoryService
- BackOfficeService
- ObjectService
- DocumentService
- RelationshipService
- NavigationService
- MultifilingService
- ManagementService
- OfficialBookService
- SubjectRegistryService
- SMSService

Nei successivi capitoli saranno illustrati le operazioni esposte per i diversi servizi e, per ognuna di queste, sono riportate le tabelle che descrivono i dati dal punto di vista logico/funzionale, per il dettaglio completo delle strutture e una spiegazione dettagliata delle operazioni esposte dai servizi, vedi i documenti in elenco al paragrafo 1.3.

### 5.1 Modalità di esposizione dei servizi ACARIS

I servizi applicativi ACARIS prevedono la modalità Web Service come principale modalità di esposizione. Il singolo servizio viene esposto tramite un Wrapper costruito davanti alla componente software che effettua l'implementazione vera e propria del servizio.

I Web Services implementati sono standard e conformi al **WS-I Basic Profile**.

Il framework utilizzato è il **CXF**.

### 5.2 Fruizione internet APIStore

Le API pubblicate sull' API Manager sono disponibili sullo Store al seguente indirizzo:

<https://api-piemonte-store.csi.it/>

Lo store espone l'elenco delle API, per accedere ai dettagli tecnici occorre seguire i passi descritti in seguito.

Si premette che l'accreditamento al portale e alla successiva sottoscrizione delle API è previsto per i soggetti autorizzati alla fruizione dei dati e servizi e quindi prevista una verifica e approvazione da parte del responsabile referente del servizio dell'ente di riferimento. Si suggerisce quindi di procedere dopo aver completato l'analisi archivistica come specificato nel paragrafo "Prerequisiti".

Per poter accedere alle API è necessario che il fruitore completi i seguenti passi operativi:

- **Accreditamento allo Store attraverso credenziali di tipo SPID (o altre tipologie previste):** selezione del pulsante Accedi e completamento dei successivi passi proposti. A valle del primo accesso verrà notificato l'accreditamento all'indirizzo email collegato alle credenziali. Questo passo permette l'accesso alla consultazione dei documenti tecnici delle API.

I passi successivi sono utili per avviare l'integrazione vera e propria delle API esposte, ed è quindi necessario aver completato l'analisi archivistica.

- **Creazione dell'applicazione contenitore delle API:** indicare il nome dell'applicativo concordato nell'analisi archivistica. A conclusione dell'operazione vengono rilasciate le credenziali associate all'applicazione da utilizzare nelle chiamate alle API (chiavi OAuth2).
- **Sottoscrizione dell'API di interesse:** si procede, per ciascuna delle API di interesse, a sottoscrivere l'utilizzo da parte dell'applicazione creata al passaggio precedente. Questo passaggio è soggetto ad una nuova fase di approvazione da parte del gruppo supporto\_Acaris.
- **Utilizzo delle chiavi OAuth2 per accedere all'API:** completata la fase precedente l'integrazione vera e propria dell'API.

## 6 RepositoryService

I servizi di questa classe permettono ai client di ACTA di recuperare informazioni quali:

- i repository disponibili su di un ACTA service endpoint,
- i metadati (Repository Name e Repository Identity) necessari a individuare l'Ente attestato su quel repository ed a riferire univocamente al repository nelle operazioni CRUD degli ACTA Services,
- le configurazioni applicative del repository.

### 6.1 L'operazione getRepositories

Si tratta dell'unico servizio non specifico per un singolo repository (Ente). Permette ai client di ACTA di individuare il repository su cui operare tra quelli disponibili ad un service endpoint.

La localizzazione del repository avviene per mezzo dei dati caratteristici dell'ente: codice, denominazione, etc.

Questa operazione non richiede parametri di ingresso.

Tabella 1: Dati output getRepositories

Liv	Dato	Annotazione
1	AcarisRepositoryEntryType	Elenco di oggetti che identificano un repository configurato nel service endpoint
-2	repositoryId	Identificativo ACARIS del repository
-2	repositoryName	Nome completo identificativo del repository

## 7 BackOfficeService

Insieme di servizi che consentono di ottenere informazioni sulle entità utilizzate nel modulo back office di ACTA che gestisce la struttura organizzativa degli utenti che sono abilitati all'utilizzo del repository. Offre anche dei servizi specifici del modello degli ACTA Services necessari per ottenere metadati per la navigazione e la ricerca degli oggetti attraverso l'interfacciamento a servizi.

### 7.1 L'operazione getPrincipalExt

Questa operazione consente di recuperare l'elenco degli eventuali "principal" che possono operare sul repository (Ente) di interesse per conto di una determinata applicazione client. Un principal è un token applicativo rilasciato da ACTA, ha durata temporale limitata ed è richiesto durante l'invocazione delle altre operazioni. Attraverso questo token, il sistema identifica l'applicativo client e l'utente per conto del quale è richiesta l'operazione: in questo modo, il sistema può effettuare i controlli di accesso alle risorse e sulle operazioni a cui un utente è abilitato, in base a quanto configurato nel back office di ACTA.

*Tabella 2: Dati input getPrincipalExt*

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA
1	idUtente	Codice fiscale dell'utente utilizzato dall'applicativo client
1	idAOO	Identificativo dell'AOO in cui è collocato l'utente
1	idStruttura	Identificativo della struttura organizzativa in cui è collocato l'utente
1	idNodo	Identificativo del nodo organizzativo in cui è collocato l'utente
1	clientApplicationInfo	Dati che identificano l'applicativo client
-2	appKey	Stringa alfanumerica che identifica univocamente l'applicativo client

*Tabella 3: Dati output getPrincipalExt*

Liv	Dato	Annotazione
1	PrincipalExtResponses	Elenco di oggetti contenenti dati relativi ad un utente collocato in organigramma ACTA
-2	principalExtResponse	Singolo oggetto
--3	principalId	Token applicativo, con valenza temporale, legato all'applicativo client e l'utente utilizzato per richiamare questa operazione
--3	collocazioneUtente	Identificativo della struttura organizzativa in cui è collocato l'utente
---4	aoo	Dato identificativo dell'AOO di collocazione
---4	struttura	Dato identificativo della struttura di collocazione
---4	nodo	Dato identificativo del nodo di collocazione
--3	profili	Elenco di descrizioni dei profili abilitati per questa collocazione
----4	identificatore	Identificatore del profilo
----4	codice	Codice del profilo

## 8 ObjectService

Per gli oggetti che possono essere ricondotti alle tipologie base dello standard CMIS, gli ACTA Services offrono operazioni CRUD per l'oggetto coinvolto identificato dal proprio ID (object ID), per ognuna delle operazioni elementari si distinguono:

- servizi di creazione di oggetti come createDocument, createFolder, createRelationship e restituzione dell'identificativo univoco;
- servizi di lettura delle proprietà degli oggetti e, opzionalmente, delle operazioni permesse su di essi;
- servizi di aggiornamento delle proprietà degli oggetti, per quelle a valore multiplo occorre sostituire l'intera lista di valori, le proprietà read-only non sono aggiornabili dai client;
- supporto all' "optimistic locking" per mezzo della proprietà ChangeToken: si tratta di un attributo generato dal repository, e da questo dipendente, che permette a quest'ultimo di verificare in fase di update se l'oggetto è stato aggiornato rispetto alla precedente lettura;
- servizi di creazione e lettura di content stream per mezzo dell'identificativo dell'oggetto Document di appartenenza, se un contenuto è già stato associato ad un Document al momento della creazione dello stream il contenuto è sovrascritto.

Non tutte le operazioni elencate sono sempre possibili per ogni istanza degli oggetti del modello, ciò non solo per le limitazioni imposte dai tipi di oggetti definiti nel modello ma anche per le policy di gestione dell'archivio che modificano nel tempo le operazioni possibili sulle istanze (trasferimenti da corrente a deposito, attività collaborative sui documenti ed altri processi propri del ciclo di vita dei contenuti).

Sebbene un content stream possa apparire una proprietà di un oggetto Document in realtà i servizi che restituiscono le proprietà dell'oggetto, o permettono di ricercare per mezzo di queste, non restituiscono un content stream.

### 8.1 L'operazione query

Consente di effettuare delle ricerche sugli oggetti di competenza del modulo funzionale Archive: appartengono a questo modulo gli oggetti che modellano le strutture aggregative, le classificazioni, i documenti.

*Tabella 4: Dati input query*

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS
1	target	Indicazione della tipologia di oggetto di cui effettuare la ricerca
1	filter	Filtro da applicare alla ricerca (oggetto di tipo PropertyFilterType, vd Tabella )
1	criteria	Elenco di criteri di ricerca
-2	criterio	Singolo criterio (oggetto di tipo QueryConditionType, vd )
1	navigationLimits	Opzionale, dati per limitare la ricerca ad un sottoalbero dei dati presenti per l'ente
1	maxItems	Valori da utilizzare per effettuare una paginazione dei risultati. <ul style="list-style-type: none"> <li>• maxItems indica il numero massimo di oggetti</li> </ul>
	skipCount	

- skipCount indica l'elemento da cui continuare nella restituzione degli oggetti trovati

*Tabella 5: Dati output query*

Liv	Dato	Annotazione
1	pagingResponse	Risultato della ricerca (oggetto di tipo PagingResponseType, vd Tabella )

### 8.2 L'operazione updateProperties

L'operazione consente l'aggiornamento delle property di un oggetto specifico.

La tipologia dell'oggetto da modificare viene stabilita facendo riferimento a quanto specificato nel parametro del tipo PropertyType.

Vengono modificate solo le properties espressamente elencate nel parametro del tipo PropertyType[[]].

La politica adottata per l'aggiornamento è quella della completa sovrascrittura dei valori esistenti per gli oggetti presenti in ACTA.

*Tabella 6: Dati input updateProperties*

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA
1	objectId	Identificativo ACARIS dell'oggetto di cui vuole aggiornare le property
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS
1	changeToken	Informazione temporale necessaria per verificare l'accesso in concorrenza.
1	properties	Elenco di properties da modificare e relativo valore (oggetti di tipo PropertyType, vd Tabella )

*Tabella 7: Dati output updateProperties*

Liv	Dato	Annotazione
1	simpleResponse	Dati relativi all'oggetto su cui è stata effettuata la modifica (oggetto di tipo SimpleResponseType, vd Tabella )

### 8.3 L'operazione moveDocument

Consente di effettuare lo spostamento di un documento (con gli eventuali allegati), da una struttura aggregativa di partenza ad un'altra di destinazione.

*Tabella 8: Dati input moveDocument*

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS
1	associativeObjectId	Identificativo ACARIS dell'oggetto da spostare
1	sourceFolderId	Identificativo ACARIS della struttura aggregativa di partenza
1	targetFolderId	Identificativo ACARIS della struttura aggregativa di destinazione

1	associativeProperties	Proprietà da indicare per pilotare l'esecuzione dell'operazione
-2	offlineMoveRequest	Indica se si vuole chiedere l'esecuzione asincrona dell'operazione qualora il documento da spostare abbia un numero di allegati superiore a quelli previsti

*Tabella 9: Dati output moveDocument*

<b>Liv</b>	<b>Dato</b>	<b>Annotazione</b>
1	objectId	Questo oggetto può assumere più valori, in alternativa tra loro <ul style="list-style-type: none"><li>• Identificativo ACARIS della richiesta asincrona</li><li>• Identificativo ACARIS dell'oggetto dopo lo spostamento, nel caso di operazione non asincrona</li></ul>

## 9 DocumentService

A fianco degli ACTA Services raccolti negli Object Services, sono messi a disposizione delle ECM application dei servizi appositamente creati per riunificare, in ottica di ottimizzazione, alcune delle operazioni di navigazione e creazione previste per la gestione del documento. Document Services mette a disposizione una versione alternativa delle operazioni contenute nel servizio Object Services: operazioni a grana “grossa” che effettuano, al loro interno, quelle operazioni di “orchestrazione” altrimenti demandate alle applicazioni client. Si tratta di “orchestrazione di funzionalità” e non di servizi: non vengono utilizzati direttamente i servizi applicativi ora disponibili ma le logiche di più basso livello da essi utilizzate.

### 9.1 L’operazione trasformaDocumentoPlaceholderInDocumentoElettronico

Consente di trasformare un documento nato come placeholder in documento elettronico: sostanzialmente siamo in presenza di un documento ACTA creato senza un documento elettronico associato.

L’operazione permette di inserire uno o più contenuti associati ad un file, con o senza firma, in base alle composizioni permesse (file singolo, con file o marca detached, XML e foglio di stile associato), dando indicazioni sull’eventuale operazione di sbustamento del file firmato.

*Tabella 10: Dati input trasformaDocumentoPlaceholderInDocumentoElettronico*

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS
1	classificazioneId	Identificativo ACARIS della classificazione “placeholder” di cui si richiede la trasformazione
1	registrazioneId	Nel caso in cui la classificazione sia protocollata, rappresenta l’identificativo ACARIS della registrazione di protocollo
1	infoRichiesta	Dati inerenti la richiesta
-2	tipoDocFisicoId	Identificativo del tipo di documento fisico
-2	composizioneId	Identificazione della composizione
-2	multiplo	Indica se il documento fisico è multiplo
-2	diventaElettronico	Valorizzare a TRUE se si vuole che il documento diventi puramente elettronico
-2	statoDiEfficaciaId	Identificativo dello stato di efficacia del documento elettronico
-2	rimandareOperazioneSbustamento	Nel caso in put sia fornito un file contenente delle informazioni di firma elettronica o marca temporale, indica se effettuare lo sbustamento o rimandare l’operazione
1	documentiFisici	Dati del documento elettronico da creare
-2	documentoFisico	Singolo documento fisico (oggetto di tipo DocumentoFisicoIRC, vd Tabella 12 )

*Tabella 11: Dati output trasformaDocumentoPlaceholderInDocumentoElettronico*

Liv	Dato	Annotazione
1	IdentificazioniTrasformazione	Dati relativi ai documenti inseriti (uno per documento)

-2	identificazione	Singolo documento
--3	relazione	Dati che indicano la relazione tra il documento ed i contenuti fisici inseriti (oggetto di tipo RelationshipPropertiesType, vd Tabella 14)
--3	dataUltimoAggiornamento	Data di ultimo aggiornamento del documento inserito
--3	failedSteps	Nel caso in cui si siano verificati errori nella verifica della firma di uno dei file inseriti, indicazione del singolo passo che è andato in errore
---4	failedStep	Passo (oggetto di tipo FailedStepInfo, vd Tabella )

## 9.2 Oggetti comuni

Di seguito sono riportate le indicazioni degli oggetti comuni a più operazioni dei Document services.

*Tabella 12: DocumentoFisicoIRC*

Liv	Dato	Annotazione
1	documentoArchivistico	Identificativo ACARIS del documento presente in ACTA
1	propertiesDocumentoFisico	Insieme di dati del documento elettronico (oggetto di tipo DocumentoFisicoPropertiesType, vd Tabella)
1	contenutiFisici	Elenco di oggetti che descrivono un contenuto fisico
-2	contenutoFisico	Singolo contenuto fisico (oggetto di tipo ContenutoFisicoIRC, vd Tabella 13)
1	azioniVerificaFirma	Elenco di oggetti per indicare come comportarsi se si presenta un errore in fase di verifica della firma del file in input
-2	azione	Singola azione per una delle fasi della verifica della firma (oggetto di tipo StepErrorAction, vd Tabella )

*Tabella 13: ContenutoFisicoIRC*

Liv	Dato	Annotazione
1	documentoFisico	Identificativo ACARIS del documento presente in ACTA
1	dataUltimoAggiornamentoDocumento	Data di ultimo aggiornamento del documento padre ACTA associato
1	propertiesContenutoFisico	Proprietà che descrivono un contenuto fisico (oggetto di tipo ContenutoFisicoPropertiesType, vd Tabella)
1	tipo	Tipo del contenuto fisico (oggetto tipo EnumStreamId)
1	stream	Dati relativi al file da associare al contenuto fisico, compreso il contenuto binario (oggetto di tipo AcarisContentStreamType, vd Tabella )
1	azioniVerificaFirma	Elenco di oggetti per indicare come comportarsi se si presenta un errore in fase di verifica della firma del file in input
-2	azione	Singola azione per una delle fasi della verifica della firma (oggetto di tipo StepErrorAction, vd Tabella )

*Tabella 14: StepErrorAction*

<b>Liv</b>	<b>Dato</b>	<b>Annotazione</b>
1	step	Indicazione del passo a cui si riferisce l'oggetto
1	action	Indicazione dell'azione da eseguire (oggetto di tipo EnumStepErrorAction)

*Tabella 15: FailedStepInfo*

<b>Liv</b>	<b>Dato</b>	<b>Annotazione</b>
1	fileName	Nome file per cui la verifica firma ha sollevato degli errori
1	failedSteps	Indicazione numerica del passo in errore

*Tabella 146: RelationshipPropertiesType*

<b>Liv</b>	<b>Dato</b>	<b>Annotazione</b>
1	relationType	Tipo di relazione presente tra gli oggetti di questa relazione
1	sourceId	Identificativo ACARIS dell'oggetto padre della relazione
1	targetId	Identificativo ACARIS dell'oggetto figlio della relazione
1	sourceType	Tipo di oggetto padre
1	targetType	Tipo di oggetto figlio

## 10 NavigationService

Per gli oggetti della classe base Folder, oltre agli object services di cui sopra, sono definiti servizi di navigazione della gerarchia di oggetti del repository.

Sono disponibili funzionalità di paging, utili per scorrere folder di dimensioni considerevoli, e di attraversamento di un sotto albero sino al livello di profondità richiesto, lo schema di attraversamento non è definibile dal client.

Anche per gli oggetti della classe base Relationship sono definiti servizi di navigazione, questi permettono di individuare tutte le relazioni, o loro sottoinsieme, aventi un determinato oggetto come source o target.

### 10.1 L'operazione getChildren

Ottiene la lista degli oggetti 'figli' contenuti in uno specifico folder (le relazioni navigabili tramite questa operazione sono descritte in [5]). Sono restituite solo le proprietà selezionate per ogni oggetto restituito. Questa operazione non restituisce l'eventuale contenuto binario associato al documento.

*Tabella 157: Dati Input getChildren*

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA
1	folderId	Identificativo ACARIS dell'oggetto di cui si richiede il figlio
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS
1	filter	Filtro da applicare alla ricerca (oggetto di tipo PropertyFilterType, vd Tabella )
1	maxItems skipCount	Valori da utilizzare per effettuare una paginazione dei risultati. <ul style="list-style-type: none"> <li>• maxItems indica il numero massimo di oggetti</li> <li>• skipCount indica l'elemento da cui continuare nella restituzione degli oggetti trovati</li> </ul>

*Tabella 18: Dati output getChildren*

Liv	Dato	Annotazione
1	pagingResponse	Risultato della ricerca (oggetto di tipo PagingResponseType, vd Tabella )

## 11 RelationshipService

Oltre i servizi raccolti nei Navigation Services, è messo a disposizione un servizio tramite che permette la navigazione dei legami tra un oggetto del classe DocumentoFisico e un oggetto della classe Document, necessario per ottenere i riferimenti necessari alla consultazione tramite ACTA Services dei content stream archiviati nel repository

### 11.1 L'operazione getObjectRelationship

Il servizio restituisce una lista di relazioni associate all'oggetto, della tipologia specificata con

riferimento alla direzione specificata.

*Tabella 19: Dati Input getObjectRelationship*

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA
1	objectId	Identificativo dell'oggetto con cui la relazione è associata
1	typeId	Tipo Relazione, viene selezionata dal seguente elenco: <ul style="list-style-type: none"> <li>• HistoryModificheTecnichePropertiesType</li> <li>• DocumentCompositionPropertiesType</li> <li>• HistoryVecchieVersioniPropertiesType</li> <li>• DocumentAssociationPropertiesType (quando ci sarà clips metallica e gruppo allegati)</li> </ul>
1	direction	Tipo di direzione source o target. I valori possibili sono: <ul style="list-style-type: none"> <li>• Source</li> <li>• Target</li> <li>• Either</li> </ul>
1	principal	Identificativo dell' attore che esegue l'operazione
1	Filter	Elenco delle properties

Interfaccia di output

response: Rappresenta la lista delle relazioni trovate con le relative properties.

## 12 OfficialBookService

Gli ACTA Services mettono a disposizione delle ECM application, un insieme di servizi per interagire con oggetti propri del modulo di gestione della protocollazione: creazione delle registrazioni di protocollo, navigazione dei legami tra gli oggetti del modulo e di recupero del dettaglio delle singole entità.

### 12.1 L'operazione creaRegistrazione

L'operazione consente la creazione di una registrazione di protocollo.

Le tipologie di registrazione a disposizione sono:

1. registrazione in arrivo
2. registrazione in partenza
3. registrazione interna

*Tabella 169: Dati Input creaRegistrazione*

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS
1	tipologiaCreazione	Indicazione del metodo di creazione da utilizzare <ul style="list-style-type: none"> <li>• Protocollazione</li> <li>• ProtocollazioneDocumentoEsistente</li> </ul>

		<ul style="list-style-type: none"> <li>• <b>ProtocollazioneDaSmistamento</b></li> </ul>
1	<b>infoRichiestaCreazione</b>	Dati per la creazione della registrazione di protocollo
-2	<b>registroId</b>	Indicazione del contenitore in cui creare la registrazione: <ul style="list-style-type: none"> <li>• <b>registroId</b> per indicazione puntuale del registro</li> <li>• <b>aooprocollanteId</b> per utilizzare il registro legato ad una specifica AOO</li> </ul>
	<b>aooprocollanteId</b>	
-2	<b>senzaCreazioneSoggettiEsterni</b>	Indica la volontà di non creare i soggetti ACTA per i corrispondenti esterni

*Tabella 20: Dati di output creaRegistrazione*

Liv	Dato	Annotazione
1	<b>IdentificazioneRegistrazione</b>	
-2	<b>registrazioneId</b>	Identificativo ACARIS della registrazione creata
-2	<b>numero</b>	Numero della registrazione creata
-2	<b>folderId</b>	Identificativo ACARIS e tipologia del folder in cui è presente la classificazione legata alla registrazione
	<b>folderType</b>	
-2	<b>dataUltimoAggiornamento</b>	Data di creazione della registrazione
-2	<b>classificazioneId</b>	Identificativo ACARIS della classificazione legata alla registrazione

Il dato di input **infoRichiestaCreazione** di tipo **RegistrazioneRequest** è utilizzato per veicolare i dati necessari per la creazione della registrazione. Questo è specializzato da sottoclassi specifiche per il metodo di creazione da utilizzare: **Protocollazione**, **ProtocollazioneDocumentoEsistente**, **ProtocollazioneDaSmistamento**.

*Tabella 21: Protocollazione*

Liv	Dato	Annotazione
1	<b>folderId</b>	Opzionale. Identificativo ACARIS oppure tipologia del folder in cui inserire la classificazione creata per la registrazione
	<b>folderType</b>	
1	<b>registrazioneAPI</b>	Metadati della registrazione da creare

*Tabella 22: ProtocollazioneDocumentoEsistente*

Liv	Dato	Annotazione
1	<b>classificazioneId</b>	Identificativo ACARIS della classificazione già presente in ACTA
1	<b>registrazioneAPI</b>	Metadati della registrazione da creare

*Tabella 23: ProtocollazioneDaSmistamento*

Liv	Dato	Annotazione
1	<b>smistamentoId</b>	Identificativo ACARIS dello smistamento da collegare alla registrazione
1	<b>registrazioneAPI</b>	Metadati della registrazione da creare

*Tabella 17: RegistrazioneAPI*

Liv	Dato	Annotazione
1	<b>tipoRegistrazione</b>	Tipo di registrazione da creare

		<ul style="list-style-type: none"> <li>• arrivo</li> <li>• partenza</li> <li>• interna</li> </ul>
1	infoCreazione	Metadati
-2	protocollante	Dati per identificare il protocollante
--3	strutturaId nodoId	Identificativo del nodo e della struttura in organigramma ACTA a cui sarà legata la registrazione
-2	numeroRegistrazionePrecedente annoRegistrazionePrecedente	In caso la registrazione sia legata ad una registrazione precedente, indicare numero ed anno
-2	registrazioneRiservata	Indica se la registrazione è riservata
-2	oggetto	Oggetto della registrazione. Utilizzato anche come oggetto della classificazione creata
-2	collocazioneCartacea	Collocazione cartacea della registrazione
-2	mezzoTrasmissivoId	Identificativo del mezzo trasmissivo
-2	documentoRiservato	Indica se il documento è riservato
-2	classificazioneProposta	Stringa che indica l'indice archivistico della voce in cui si richiede la classificazione.
-2	dataDocumento	Data del documento della registrazione
-2	descrizioneAllegato	Opzionale. Elenco di descrizioni degli allegati, nel caso di creazione della classificazione (metodo: Protocollazione)
--3	descrizione	Descrizione dell'allegato usata come oggetto dell'allegato
-2	mittentiInterni destinatariInterni destinatariEsterni	Elenco dei corrispondenti della registrazione di protocollo.
--3	mittenteInterno destinatarioInterno destinatarioEsterno	Singolo mittente/destinatario
---	corrispondente	Dati del corrispondente (oggetto di tipo InfoCreazioneCorrispondente)
4		
-2	annotazione	Annotazione da inserire sulla registrazione
-2	forzareSeRegistrazioneSimile	Indica se effettuare la creazione in caso di presenza di una registrazione simile
-2	forzareSeRegistrazionePropostaInvalida	Indica se effettuare la creazione in caso di errore sulla indicazione della classificazione proposta
-2	forzareSePresenzaInviti	Indica se effettuare la creazione di inviti automatici in lettura sulla registrazione
-2	forzareSePresenzaDaInoltrare	Indica se effettuare la creazione in caso di presenza di inoltri verso altra AOO

Il dato di input di tipo RegistrazioneAPI indica il tipo di registrazione di protocollo di cui è richiesta la creazione (tra Arrivo, Partenza e Interna). Questo è specializzato dalle sottoclassi: RegistrazioneArrivo, RegistrazionePartenza, RegistrazioneInterna. La prima di queste aggiunge alcuni dati supplementare rispetto il tipo base.

*Tabella 18: RegistrazioneArrivo*

Liv	Dato	Annotazione
1	infoDateArrivo	Informazioni sulla data di arrivo della registrazione
-2	dataRicezione	Data di ricezione della registrazione

-2	dataTimbroPostale	Data del timbro postale
1	infoProtocolloMittente	Se la registrazione è legata ad una registrazione del mittente, indica i dati di quella registrazione
-2	numero	Numero della registrazione mittente
-2	anno	Anno della registrazione mittente
-2	data	Data della registrazione mittente
1	mittentiEsterni	Elenco di mittenti esterni
-2	mittenteEsterno	Mittente
--3	corrispondente	Dati del corrispondente (oggetto di tipo InfoCreazioneCorrispondente)

### 13 MultifilingServices

Per gli oggetti della classe base Folder, oltre agli Object Services di cui sopra, sono definiti servizi di mantenimento della gerarchia di oggetti del repository. Gli ACTA Services permettono di creare associazioni di un oggetto derivato dalla classe base Document ad uno o più oggetti (si supporta il multi-filing) derivati dalla classe base Folder.

Invece gli oggetti derivati dalla classe base Folder non possono essere associati a più di un Folder, causa il vincolo che impone di avere un solo padre, e ad un oggetto della propria discendenza per il vincolo di ciclicità dell'albero, per questo è stato previsto un servizio specifico.

#### 13.1 L'operazione addAssociativeObjectToFolder

Aggiunge un oggetto di tipo Document ad un folder creando il folder 'associativo' di tipo classificazione, come folder può essere solo specificato un oggetto di tipo:

- ✓ DossierPropertiesType
- ✓ SottofascicoloPropertiesType
- ✓ FascicoloRealeEreditatoPropertiesType
- ✓ FascicoloRealeLiberoPropertiesType
- ✓ FascicoloRealeContinuoPropertiesType
- ✓ FascicoloRealeLegislaturaPropertiesType
- ✓ FascicoloRealeAnnualePropertiesType
- ✓ VolumeSerieTipDocPropertiesType
- ✓ SerieTipologicaDocumentiPropertiesType
- ✓ VolumeSottoFascPropertiesType
- ✓ VolumeFascPropertiesType
- ✓ ClassificazionePropertiesType

come folder associativo può essere specificato solo:

- ✓ ClassificazionePropertiesType

mentre come document può essere specificato solo un oggetto di tipo :

- ✓ DocumentoDBPropertiesType
- ✓ DocumentoRegistroPropertiesType
- ✓ DocumentoSemplicePropertiesType

Se vengono specificati degli oggetti non validi il sistema genera una AcarisException con il messaggio SERMUL-E001

*Tabella 19: Dati di input addAssociativeObjectToFolder*

Liv	Dato	Annotazione
1	repositoryId	Repository Id
1	objectId	Documento da aggiungere al folder. I tipi di oggetto ammessi sono: <ul style="list-style-type: none"> <li>✓ DocumentoDBPropertiesType</li> <li>✓ DocumentoRegistroPropertiesType</li> <li>✓ DocumentoSemplicePropertiesType</li> </ul>
1	Principal	Identifica univocamente l'attore che vuole operare sul repository, questo deve essere presente su ACTA ed autorizzato ad operare sui contenuti, viene verificata l'autorizzazione alla creazione dello specifico oggetto Document richiesto (abilitazione al corrispettivo CdU in ACTA).
1	folderId	Folder che conterrà il folder 'associativo'. I tipi di oggetto ammessi sono: <ul style="list-style-type: none"> <li>✓ DossierPropertiesType</li> <li>✓ SottofascicoloPropertiesType</li> <li>✓ FascicoloRealeEreditatoPropertiesType</li> <li>✓ FascicoloRealeLiberoPropertiesType</li> <li>✓ FascicoloRealeContinuoPropertiesType</li> <li>✓ FascicoloRealeLegislaturaPropertiesType</li> <li>✓ FascicoloRealeAnnualePropertiesType</li> <li>✓ VolumeSerieTipDocPropertiesType</li> <li>✓ SerieTipologicaDocumentiPropertiesType</li> <li>✓ VolumeSottoFascPropertiesType</li> <li>✓ VolumeFascPropertiesType</li> <li>✓ ClassificazionePropertiesType</li> </ul>
1	AssociationPropertiesType	I tipi di oggetto ammessi sono: ClassificazionePropertiesType
1	propertiesAssociazione	Elenco delle properties che caratterizzano il documento

### 13.2 L'operazione aggiungiClassificazione

Il servizio costituisce un'evoluzione del servizio addAssociativeObjectToFolder: serve allo stesso scopo (multi classificazione di un documento) e quindi implementa le stesse logiche di business ma introduce alcune innovazioni in merito alla gestione degli allegati, alla possibilità di prenotare una copia offline degli allegati e per quanto riguarda il valore ritornato come output.

*Tabella 26: Dati di input aggiungiClassificazione*

Liv	Dato	Annotazione
-----	------	-------------

1	repositoryId	Repository Id
1	Principal	Identifica univocamente l'attore che vuole operare sul repository, questo deve essere presente su ACTA ed autorizzato ad operare sui contenuti, viene verificata l'autorizzazione alla creazione dello specifico oggetto Document richiesto (abilitazione al corrispettivo CdU in ACTA).
1	classificazione di partenza del documento da copiare	I tipi di oggetto ammessi sono: ✓ Classificazione
1	folderId	Folder che conterrà il folder 'associativo'. I tipi di oggetto ammessi sono: ✓ DossierPropertiesType ✓ SottofascicoloPropertiesType ✓ FascicoloRealeEreditatoPropertiesType ✓ FascicoloRealeLiberoPropertiesType ✓ FascicoloRealeContinuoPropertiesType ✓ FascicoloRealeLegislaturaPropertiesType ✓ FascicoloRealeAnnualePropertiesType ✓ VolumeSerieTipDocPropertiesType ✓ SerieTipologicaDocumentiPropertiesType ✓ VolumeSottoFascPropertiesType ✓ VolumeFascPropertiesType ✓ ClassificazionePropertiesType
1	Collection params	Elenco di oggetti di tipo ItemType veicolati tramite un oggetto di tipo VarargsType da utilizzare come indicazioni per specificare il tipo di operazione da effettuare. Nell'attuale versione del servizio i parametri che possono essere specificati sono: ✓ addConAllegati ✓ offlineAddRequest ✓ numeroClassificazione

Indicazioni in merito all'utilizzo dei parametri previsti da Varargs (Collection params)

### **addConAllegati**

se "false" viene aggiunta una classificazione semplice senza gruppo allegati

se "true" viene aggiunta una classificazione con gruppo allegati e se ci sono allegati questi vengono copiati

### **offlineAddRequest**

il funzionamento è simile a quello di moveDocument

se false, se addConAllegati è true e se il sistema verifica che il numero degli allegati è superiore a quello ammesso viene sollevata un'eccezione

se true, se addConAllegati è true e se il sistema verifica che il numero degli allegati è superiore a quello ammesso il sistema salva una prenotazione dell'esecuzione del batch di copia e incolla degli allegati

### **numeroClassificazione**

la valorizzazione è consentita solo in alcuni casi: negli altri è il sistema che se ne occupa.

Comportamento del sistema nel caso di inserimento in serie tipologica di documenti:

Serie tipologica a numerazione libera:

- numeroClassificazione è una stringa, obbligatoria nel caso di inserimento in una serie(o volume) a numerazione libera; il sistema valorizzerà il campo codice della classificazione;

Serie tipologica altre numerazioni:

- numeroClassificazione è una stringa, in tutte le numerazioni automatiche, che venga inserito o meno verrà in ogni caso calcolato in automatico;

Quindi, nei soli casi in cui è consentito, bisogna valorizzare il campo numeroInput con un valore a scelta del fruitore.

I valori dei parametri in oggetto NON sono considerati *case sensitive*.

Riepilogo del flusso di chiamate:

- se aggiungo una classificazione senza allegati mi viene restituito l'identificatore della classificazione
- se aggiungo una classificazione con allegati e richiedo l'esecuzione offline, se il sistema rileva che può essere effettuata una prenotazione, restituisce l'identificatore della prenotazione
- se si chiede l'esecuzione offline, per capire se l'identificatore restituito è una prenotazione bisogna invocare query con target moveDocumentOfflineView: se viene restituita eccezione SERQRY-E007 allora l'identificatore passato in input non è una prenotazione

Quando la classificazione richiesta ha come folderId un oggetto di tipo ClassificazionePropertiesType, questo significa che si vuole creare un allegato ad altro documento, è necessario che il documento da allegare non abbia a sua volta allegati: in caso contrario il sistema restituisce un'eccezione veicolando il codice di errore SER\_E\_173.

### 13.1.1 Output

ObjectId rappresentante la classificazione creata oppure, in caso di prenotazione di un'operazione di copia offline degli allegati, della prenotazione dell'esecuzione del batch. Nel caso di prenotazione lo stato della stessa può essere verificata tramite la vista "moveDocumentOfflineView": segue la struttura della stessa. L'eventuale invocazione della vista con l'objectId della classificazione (in luogo di quello della prenotazione) causa la restituzione di un'eccezione Acaris con codice SERQRY-E007 e messaggio "Uno o più dei criteri indicati non sono utilizzabili come parametro di ricerca nel servizio di query".

I valori relativi ai diversi stati di elaborazione sono recuperabili tramite la vista "StatoElaborazioneBatchDecodifica".

## 14 ManagementService

Gli ACTA Services permettono di operare su oggetti del modello concettuale di ACTA ed in aggiunta su entità di business non direttamente sotto il controllo di tale modello ma comunque necessarie per l'ERMS. Si tratta delle entità del modulo BKO di ACTA richieste in fase di creazione di oggetti derivati da Document

### 14.1 L'operazione getVitalRecordCode

Restituisce l'elenco dei VitalRecordCode di un Repository (Ente)

*Tabella 27: Dati di input aggiungiClassificazione*

Liv	Dato	Annotazione
1	IdRepository	Repository Id

#### 14.1.1.1 Interfaccia di output

Elenco di oggetti contenente le property di ogni vital record code trovato:

- idVitalrecordCode: identificativo univoco del vital record code del repository
- descrizione : stringa contenente la descrizione del vital record code
- tempoDiVitalità: tempo di vitalità in giorni del vitl record code

### 14.2 L'operazione query

*Tabella 28: Dati di input aggiungiClassificazione*

Liv	Dato	Annotazione
1	IdRepository	Repository Id
1	principalId	Identifica univocamente l'attore che vuole operare sul repository
1	Target	<b>QueryableObjectType:</b> Rappresenta l'entità interrogabile. Può essere un'entità fisica oppure una vista dinamica
1	filter	<b>PropertyFilterType:</b> Permette di specificare l'elenco delle properties che devono essere restituite per ogni oggetto individuato nel sistema.
1	criteria	<b>QueryConditionType[]:</b> Elenco di criteri (in questa prima versione concatenate in AND) di selezione
1	maxItems	<b>Integer</b>
1	skipCount	<b>Integer:</b> Vengono utilizzati per gestire la paginazione dei risultati

## 15 SubjectRegistryService

I servizi raccolti nei Subject Registry Services permettono di agire su oggetti delle classi specifiche di ACTA del modulo di gestione del protocollo: creazione di soggetti gestiti nell'anagrafica dei corrispondenti, loro aggiornamento e recupero delle informazioni di questa tipologia di oggetti.

### 15.1 L'operazione creaSoggetto

L'operazione consente la creazione di un Soggetto applicando, in quanto compatibili con una esposizione a servizi, le logiche già previste per questa funzionalità nel modulo PRT di Acta

*Tabella 29: Dati di input creaSoggetto*

Liv	Dato	Annotazione
1	IdRepository	Repository Id
1	principallId	Identifica univocamente l'attore che vuole operare sul repository
1	tipologiaCreazione	<b>EnumTipologiaCreazioneSoggetto:</b> il fruitore indica la tipologia di soggetto da creare: <i>SoggettoProvvisorio</i> o <i>SoggettoDefinitivo</i> .
1	infoRichiestaCreazione	<b>SoggettoRequest:</b> Individuato il tipo di soggetto da creare il parametro <i>infoRichiestaCreazione</i> consente di specificare, conformemente alle indicazioni contenute in <i>tipologiaCreazione</i> , le informazioni specifiche del soggetto da creare. Il parametro <i>infoRichiestaCreazione</i> è astratto e quindi ci si aspetta che venga valorizzato con una specifica sottoclasse dello stesso. In caso contrario il sistema solleva apposita eccezione.

### 15.1.1 Output

Se il soggetto viene correttamente creato, il sistema restituisce un oggetto di tipo *IdentificazioneSoggetto* che è così definito:

```
<xs:complexType name="IdentificazioneSoggetto">
  <xs:sequence>
    <!-- objectName:propertyName -->
    <xs:element name="soggettoId" type="common:ObjectIdType"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="tipoSoggetto" type="tns:enumTipologiaCreazioneSoggetto"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="dataUltimoAggiornamento" type="common:ChangeTokenType"
      minOccurs="1" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>
```

L'oggetto permette di individuare l'identificatore del soggetto creato, se si tratta di soggetto definitivo o provvisorio e un oggetto di tipo *ChangeToken* che può essere utilizzato in eventuali operazioni di aggiornamento.

## 15.2 L'operazione *getProperties*

Operazione di interrogazione puntuale degli oggetti del modulo *SubjectRegistry* di tipo *Properties*.

*Tabella 30: Dati di input *getProperties**

Liv	Dato	Annotazione
1	IdRepository	Repository Id
1	principallId	Identifica univocamente l'attore che vuole operare sul repository
1	objectId	<b>ObjectIdType :</b> Identificatore di cui restituire il dettaglio
1	filter	<b>PropertyFilterType :</b> Proprietà da valorizzare per il nodo interrogato

### 15.2.1 Output

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

### 15.3 L'operazione updateProperties

L'operazione consente l'aggiornamento delle properties di un oggetto specifico del modulo SubjectRegistry: il sistema verifica che si tratti di un oggetto di tipo properties del modulo indicato e solleva apposita eccezione nel caso di verifica negativa.

Vengono modificate solo le properties espressamente elencate nel parametro *properties* di tipo PropertyType[].

La politica adottata per l'aggiornamento è quella della completa sovrascrittura dei valori esistenti nel repository.

Il servizio di updateProperties, non deve essere utilizzato per aggiornare i campi che determinano la modifica dello stato dell'entità in quanto tale operazione potrebbe coinvolgerne altre ad alto livello di complessità e non necessariamente gestibili in unica operazione. In questi casi, la modifica viene gestita tramite appositi e indipendenti casi d'uso.

Per evitare che questi metadati vengano comunque modificati tramite updateProperties, anche tramite involontaria impostazione a true dell'apposito flag, il sistema effettua una verifica puntuale che il metadato non sia presente nell'apposita lista di metadati delicati.

A questo proposito il sistema deve controllare che la singola property che si intende aggiornare non sia compresa nella tabella *acta\_c\_proprietà\_non\_agg* e, se necessario, solleva apposita eccezione.

*Tabella 31: Dati di input updateProperties*

Liv	Dato	Annotazione
1	IdRepository	Repository Id
1	principalId	Identifica univocamente l'attore che vuole operare sul repository
1	objectId	<b>ObjectIdType</b> : Identificatore di cui restituire il dettaglio
1	changeToken	<b>ChangeTokenType</b> : informazione necessaria per il controllo della concorrenza
1	properties	<b>PropertyType[]</b> : elenco delle properties da modificare. Il sistema sovrascrive i soli metadati specificati nell'elenco. La stessa politica viene adottata nel caso di metadati multivalore

#### 15.3.1 Output

Viene restituito un oggetto di tipo *SimpleResponseType*.

Oggetto complesso costituito dall'identificatore dell'oggetto modificato e dal changeToken relativo alla modifica effettuata.

### 15.4 L'operazione query

Consente di effettuare delle query sul modulo funzionale Acaris SubjectRegistry.

Sono interrogabili sono gli oggetti che il sistema considera appartenenti al modulo SubjectRegistry.

Si ricorda che il servizio viene esposto dai singoli moduli Acaris allo scopo di consentire, all'interno di un modulo, l'interrogazione delle sole entità afferenti al modulo considerato; allo stesso tempo, però, le logiche da applicare sono comuni a tutti i moduli e quindi anche le verifiche che il sistema effettua.

*Tabella 32: Dati di input updateProperties*

Liv	Dato	Annotazione
1	IdRepository	Repository Id

1	<b>principalId</b>	Identifica univocamente l'attore che vuole operare sul repository
1	<b>target</b>	<b>QueryableObjectType</b> : Rappresenta l'entità interrogabile. Può essere un'entità fisica oppure una vista dinamica.
1	<b>filter</b>	<b>PropertyFilterType</b> : Permette di specificare l'elenco delle properties che devono essere restituite per ogni oggetto individuato nel sistema.
1	<b>criteria</b>	<b>QueryConditionType[]</b> :Elenco di criteri (in questa prima versione concatenate in AND) di selezione
1	<b>maxItems</b>	Viene utilizzato per gestire la paginazione dei risultati
1	<b>skipCount</b>	Viene utilizzato per gestire la paginazione dei risultati

#### 15.4.1 Output

Viene restituito un oggetto di tipo PagingResponseType. In sostanza si tratta di un oggetto che consente di gestire la paginazione di una serie di risultati rappresentati come identificatore e collection di properties

#### 15.5 L'operazione ricercaSoggetto

Si tratta di una figura particolare di query che, in primo luogo cerca sulla fonte esterna indicata dal fruitore e, in caso di mancanza di risultati, ripete la stessa sulle fonti interne tramite il normale impiego del servizio query.

Tabella 33: Dati di input *ricercaSoggetto*

Liv	Dato	Annotazione
1	<b>IdRepository</b>	Repository Id
1	<b>principalId</b>	Identifica univocamente l'attore che vuole operare sul repository
1	<b>identitaDigitale</b>	Si tratta di un'identità IRIDE ed è obbligatoria in quanto richiesta nell'interazione con il servizio che effettua la ricerca sulla fonte esterna ad ACTA.
1	<b>codiceFiscale</b>	Almeno uno tra codice fiscale e partita IVA dei due deve essere valorizzato
1	<b>partitaIva</b>	Almeno uno tra codice fiscale e partita IVA dei due deve essere valorizzato
1	<b>fonteEsterna</b>	Consente di specificare la fonte sulla quale effettuare la ricerca. Nella versione attuale del servizio si può specificare solo "GMS".
1	<b>flagPersonaGiuridica</b>	Consente di specificare se il soggetto ricercato debba essere considerato o meno come persona giuridica

Se la ricerca viene eseguita con successo il sistema restituisce un oggetto di tipo AnagraficaGenerica. Segue la struttura della classe come definita nell'apposito xml schema.

```
<xs:complexType name="AnagraficaGenerica">
  <xs:sequence>
    <!-- objectName:propertyName -->
    <xs:element name="tipoAnagrafica" type="common:string"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="codiceFonte" type="common:string"
```

```

        minOccurs="1" maxOccurs="1" />
<xs:element name="descrizioneFonte" type="common:string"
        minOccurs="1" maxOccurs="1" />
<xs:element name="chiaveEsterna" type="common:string"
        minOccurs="0" maxOccurs="1" />
<xs:element name="objectId" type="common:ObjectIdType"
        minOccurs="0" maxOccurs="1" />
<xs:element name="dbKey" type="common:IDDBType"
        minOccurs="0" maxOccurs="1" />
<xs:element name="codiceFiscale" type="common:string"
        minOccurs="0" maxOccurs="1" />
<xs:element name="nome" type="common:string"
        minOccurs="0" maxOccurs="1" />
<xs:element name="cognome" type="common:string"
        minOccurs="0" maxOccurs="1" />
<xs:element name="partitaIva" type="common:string"
        minOccurs="0" maxOccurs="1" />
<xs:element name="denominazione" type="common:string"
        minOccurs="0" maxOccurs="1" />
<xs:element name="idStatoSoggetto" type="common:IDDBType"
        minOccurs="0" maxOccurs="1" />
<xs:element name="descStatoSoggetto" type="common:string"
        minOccurs="0" maxOccurs="1" />
</xs:sequence>
</xs:complexType>

```

## 15.6 L'operazione ricercaSggettoDaFonteEsterna

Si tratta di un servizio generico di ricerca di un soggetto su fonte esterna ad ACTA.

*Tabella 34: Dati di input ricercaSoggettoDaFonteEsterna*

Liv	Dato	Annotazione
1	<b>IdRepository</b>	Repository Id
1	<b>principallId</b>	Utente che richiede il servizio
1	<b>identitaIride</b>	Si tratta di un'identità IRIDE ed è obbligatoria in quanto richiesta nell'interazione con il servizio che effettua la ricerca sulla fonte esterna ad ACTA.
1	<b>criteri</b>	<b>CriteriRicercaSoggetto</b> : i criteri necessari per indirizzare la tipologia di ricerca e il servizio esterno da invocare

### 15.6.1 Output

<Array> **AnagraficaSoggetto**: Eventuale elenco di soggetti trovati nell'anagrafica esterna

## 15.7 L'operazione getPropertiesMassive

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

*Tabella 35: Dati di input **getPropertiesMassive***

Liv	Dato	Annotazione
1	<b>IdRepository</b>	Repository Id
1	<b>principalId</b>	Utente che richiede il servizio
1	<b>identificatori</b>	<b>ObjectIdType[]</b> : Elenco di identificatori di cui restituire il dettaglio
1	<b>filter</b>	<b>PropertyFilterType</b> : Proprietà da valorizzare per il nodo interrogato

#### 15.7.1 Output

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

## 16 SMSService

### 16.1 L'operazione query

Servizio analogo a quello presente nei Object Service. Permette di accedere alle proprietà degli oggetti di competenza del modulo di gestione degli smistamenti.

*Tabella 36: Dati di input **query***

Liv	Dato	Annotazione
1	<b>IdRepository</b>	Repository Id
1	<b>principalId</b>	Identifica univocamente l'attore che vuole operare sul repository
1	<b>target</b>	<b>QueryableObjectType</b> : Rappresenta l'entità interrogabile. Può essere un'entità fisica oppure una vista dinamica.
1	<b>filter</b>	<b>PropertyFilterType</b> : Permette di specificare l'elenco delle properties che devono essere restituite per ogni oggetto individuato nel sistema.
1	<b>criteria</b>	<b>QueryConditionType[]</b> :Elenco di criteri (in questa prima versione concatenate in AND) di selezione
1	<b>maxItems</b>	Viene utilizzato per gestire la paginazione dei risultati
1	<b>skipCount</b>	Viene utilizzato per gestire la paginazione dei risultati

### 16.2 L'operazione creaSmistamento

Consente di creare un oggetto di tipo Smistamento. Un oggetto di questo tipo permette, per esempio, di poter rendere possibile l'accesso ad un Document ad utenti dell'organigramma che non ne avrebbero accesso in base alla loro collocazione o di "assegnare" la competenza di un Document ad un determinato gruppo di utenti che condividono la stessa collocazione in organigramma.

*Tabella 37: Dati di input creaSmistamento*

Liv	Dato	Annotazione
1	<b>IdRepository</b>	Repository Id
1	<b>principallId</b>	Utente che richiede il servizio
1	<b>mittente</b>	E' un utente fisico.
1	<b>Destinatari</b>	<b>Destinatario[]</b> :Può essere una lista di utenti oppure nodi. Se nodo può essere organizzativo oppure di smistamento (nel caso di inoltro). Nel caso di nodo smistamento bisogna valorizzare anche l'identificatore del nodo organizzativo. Può essere anche un destinatario per conoscenza. Se richiesto dalla tipologia di smistamento viene valorizzato l'identificatore del fascicolo temporaneo.
1	<b>oggettiSmistati</b>	<b>OggettoSmistamento[]</b> : Una lista di 1..n classificazioni: se lo smistamento parte da PRT o da fascicolo temporaneo di titolare di GCO si ha un'unica classificazione. Nel caso in cui lo smistamento abbia origine dal modulo PRT allora bisogna valorizzare, anche l'identificatore della registrazione di protocollo
1	<b>infoCreazione</b>	infoCreazione: Oggetto che contiene una serie di informazioni necessarie in fase di creazione di uno smistamento: alcuni sono valorizzati solo in caso di inoltro. <ul style="list-style-type: none"> <li>• Tipo smistamento: indica l'identificatore della tipologia di smistamento da creare. Il valore viene ricavato tramite navigazione o query puntuale sul QueryableObject di riferimento.</li> <li>✓ Visibile a tutti: indica se lo smistamento debba essere visibile a tutti gli appartenenti al nodo indicato come nodo mittente (di tipo organizzativo)</li> <li>✓ Note</li> <li>✓ Modifica doc</li> <li>✓ Modifica con versionamento</li> <li>✓ IdentificatoreSmistamento (da valorizzare solo in caso di inoltro). Serve per legare lo smistamento che si crea in seguito ad inoltro con lo SmistamentoDestinatario dell'inoltrante.</li> </ul>

### 16.2.1 Output

Il servizio restituisce l'identificatore dello smistamento creato.

Si tratta precisamente dell'identificatore dello SmistamentoMittente creato con l'operazione.

- ✓ IdSmistamento idSmistamento

## 17 Oggetti comuni a tutti i servizi ACARIS

Di seguito sono riportate le indicazioni degli oggetti comuni a più servizi esposti.

*Tabella 38: DocumentoFisicoPropertiesType*

Liv	Dato	Annotazione
1	descrizione	Descrizione del documento fisico

*Tabella 39: ContenutoFisicoPropertiesType*

Liv	Dato	Annotazione
1	contentStreamFileName	Nome del file associato al contenuto fisico
1	contentStreamLength	Dimensione (in byte) del file associato al contenuto fisico
1	contentStreamMimeType	Mime type del file associato al contenuto fisico
1	sbustamento	Indica se il contenuto fisico deve essere sbustato
1	modificabile	Indica se il file associato al contenuto fisico è di tipo modificabile

*Tabella 40: AcarisContentStreamType*

Liv	Dato	Annotazione
1	streamMTOM	Oggetto che veicola il contenuto binario del file in modalità MTOM
1	filename	Nome del file associato
1	length	Lunghezza del file associato
1	mimeType	Mime type del file associato
1	nodeUID	Identificativo del contenuto fisico presente su ACTA

*Tabella 41 QueryConditionType*

Liv	Dato	Annotazione
1	propertyName	Nome della property
1	operator	Operatore di confronto
1	value	Valore della property

*Tabella 42 NavigationConditionInfoType*

Liv	Dato	Annotazione
1	parentNodeId	Identificativo ACTA dell'oggetto che rappresenta il nodo radice dell'albero in cui effettuare la ricerca
1	limitToChildren	Indica se la ricerca deve essere effettuata per i figli diretti di parentNode o per tutta la sua discendenza

*Tabella 43: PropertyFilterType*

Liv	Dato	Annotazione
-----	------	-------------

1	filterType	Tipo di filtro da utilizzare <ul style="list-style-type: none"> <li>• ALL, nessun filtro</li> <li>• NONE, insieme minimo delle proprietà</li> <li>• LIST, restituire solo le property indicate in propertyList</li> </ul>
1	propertyList	Elenco di property di cui si richiede la restituzione
-2	property	Singola property (oggetto di tipo QueryNameType, vd Tabella )

*Tabella 44: PagingResponseType*

Liv	Dato	Annotazione
1	objects	Elenco di oggetti trovati con la ricerca
-2	object	Singolo oggetto (oggetto di tipo ObjectResponseType, vd Tabella )
--3	objectId	Identificativo ACARIS dell'oggetto
--3	properties	Elenco di property dell'oggetto restituito
---4	property	Singola property (oggetto di tipo QueryNameType, vd Tabella )
1	hasMoreItems	Nel caso di paginazione, indica la presenza di altri oggetti inerenti alla ricerca effettuata

*Tabella 45: QueryNameType*

Liv	Dato	Annotazione
1	className	Nome della classe di riferimento della property
1	propertyName	Nome della property

*Tabella 46: ObjectResponseType*

Liv	Dato	Annotazione
1	objectId	Identificativo ACARIS dell'oggetto
1	properties	Elenco di property dell'oggetto restituito (oggetto di tipo PropertyType, vd Tabella )

*Tabella 47: PropertyType*

Liv	Dato	Annotazione
1	queryName	Dati identificativi della property (oggetto di tipo QueryNameType, vd Tabella )
1	value	Valore della property

*Tabella 48: SimpleResponseType*

Liv	Dato	Annotazione
1	objectId	Identificativo ACARIS di un oggetto esistente
1	changeToken	Indicazione temporale dell'ultima modifica effettuata sull'oggetto