



Specifiche di evoluzione del sistema di protocollo informatico e di gestione documentale p@doc - gestione dell'archivio storico



Indice generale

1. OBIETTIVO DEL DOCUMENTO.....	4
2. NORME E STANDARD DI RIFERIMENTO.....	4
3. GLI STRUMENTI DELLA CONSERVAZIONE NELLA GESTIONE DOCUMENTALE.....	5
3.1 LA RICOGNIZIONE DEI PROCEDIMENTI E DEI DOCUMENTI.....	5
3.2 IL PIANO DI CLASSIFICAZIONE.....	8
3.3 ASPETTI DELLA CONSERVAZIONE NELLA FORMAZIONE DEL DOCUMENTO INFORMATICO E DEL FASCICOLO INFORMATICO.....	9
4. LA CONSERVAZIONE NEGLI ENTI DI PRODIGIO.....	14
5. TAKO - ANALISI DEL MODELLO APPLICATIVO PER IL VERSAMENTO DEI DOCUMENTI.....	15
5.1 Analisi del contesto.....	15
5.2 Concept di massima.....	17
5.3 Macro struttura SIP Parer.....	19
5.3.1 SIP/Unità documentaria.....	19
5.3.2 Documenti.....	21
5.3.3 Componente.....	24
5.3.4 Sotto-componente.....	26
5.3.5 Sintesi struttura.....	27
5.4 Schema e/r di massima.....	29
5.5 Flusso di lavoro, concept del processo.....	32
5.5.1 Processo parsing.....	34
5.5.2 Processo conversion.....	35
5.6 Parser, UnitDocObject, connettore e converter.....	37
5.6.1 UnitDocObject.....	42
5.6.2 Parser ed XMLSipIndex.....	46
5.6.3 Connuttore, converter.....	58



1. OBIETTIVO DEL DOCUMENTO

Questo documento vuole analizzare le attività di implementazione all'interno del sistema di protocollo informatico e di gestione documentale [p@doc](#) nell'ambito del potenziamento della componente di gestione dell'archivio storico documentale. Nello specifico l'analisi si focalizza nella definizione delle tipologie documentali da inviare in conservazione, nell'individuazione dei sistemi e dei modelli organizzativi di conservazione adottati dai vari partner di progetto, e nella definizione delle componenti del connettore di conservazione da implementare all'interno del sistema di protocollo informatico e di gestione documentale p@doc.

2. NORME E STANDARD DI RIFERIMENTO

Il seguente documento fa riferimento alle norme e agli standard nazionali ed internazionali in materia di amministrazione digitale e gestione documentale, tutela e conservazione degli archivi, protezione dei dati personali e sicurezza informatica. Nello specifico:

- Decreto del Presidente della Repubblica 20 dicembre 2000 n. 445 - *Testo unico delle disposizioni legislative e regolamentari in materia di documentazione amministrativa*
- Decreto legislativo 7 marzo 2005 n. 82- *Codice dell'amministrazione digitale*
- Decreto Legislativo 22 gennaio 2004, n. 42- *Codice dei beni culturali e del paesaggio*
- Decreto Legislativo 30 giugno 2003, n. 196- *Codice in materia di protezione dei dati personali*
- Decreto del Presidente del Consiglio dei Ministri 3 dicembre 2013 – *Regole tecniche in materia di sistema di conservazione*
- ISO 15489-1:2016 - Information and documentation - Records management – Part 1: Concepts and principles
- ISO 14721:2002 - OAIS (Open Archival Information System) - *Sistema informativo aperto per l'archiviazione.*
- ISO/IEC 27001:2005 - *Information technology - Security techniques - Requisiti di un ISMS (Information Security Management System).*



- ETSI TS 101 533-1 V1.1.1 (2011-05) Technical Specification, Electronic Signatures and Infrastructures (ESI); Information Preservation Systems Security - *Requisiti per realizzare e gestire sistemi sicuri e affidabili per la conservazione elettronica delle informazioni.*
- UNI 11386:2010 Standard SInCRO - *Supporto all'Interoperabilità nella Conservazione e nel Recupero degli Oggetti digitali.*

3. GLI STRUMENTI DELLA CONSERVAZIONE NELLA GESTIONE DOCUMENTALE

Quando si parla di conservazione ci si riferisce principalmente al processo attraverso il quale si garantiscono i requisiti di autenticità, integrità, affidabilità, leggibilità e reperibilità dei documenti; tuttavia, nell'applicazione del Codice dell'amministrazione digitale e delle relative linee guida, emerge la necessità di governare e di ripensare all'intero ciclo di vita del documento informatico all'interno dell'archivio digitale, nonché ai procedimenti e ai processi afferenti.

Con questo capitolo si vuole proporre un modello di analisi per identificare le caratteristiche e i requisiti normativi e tecnologici in funzione alla conservazione ma riconducibili per lo più ai processi di formazione, registrazione, classificazione e archiviazione.

3.1 LA RICOGNIZIONE DEI PROCEDIMENTI E DEI DOCUMENTI

Prima di attivare il processo di conservazione è necessario individuare e quantificare la produzione documentale al fine di prevedere costi, strategie per la produzione, gestione e conservazione dei documenti. Il modello proposto prevede in primo luogo l'analisi del procedimento amministrativo, essendo la produzione dei documenti strettamente correlata all'attività amministrativa dell'organizzazione. Data la complessità nella ricognizione dei procedimenti è consigliabile partire dalle basi informative già presenti e consolidate; in particolare è possibile fare riferimento all'obbligo di pubblicazione relativo ai procedimenti amministrativi previsto dall'articolo 35 del D.Lgs 33/2013. Inoltre, l'organizzazione dovrà rilevare tutti i procedimenti amministrativi interni che prevedono la produzione di documenti soggetti a conservazione (ad esempio quelli afferenti alle risorse umane o alle risorse finanziarie). Si costituirà in questo modo una base di dati completa e che consentirà alla stessa amministrazione di avviare un processo di semplificazione e miglioramento dei propri servizi.



Di seguito una proposta per il censimento dei procedimenti amministrativi:

Oggetto dell'analisi	Elemento	Descrizione
Procedimento amministrativo	Identificativo del procedimento	E' fondamentale associare ad ogni procedimento un ID univoco
	Nome del procedimento	Inserire la denominazione del procedimento amministrativo
	Unità Organizzativa Responsabile (UOR)	Indicare l'Unità Organizzativa Responsabile (UOR)
	Servizio o Ufficio Responsabile	Indicare il servizio o l'ufficio responsabile
	Responsabile del procedimento	Indicare il nome o il ruolo del Responsabile del procedimento amministrativo
	Descrizione del flusso	Descrivere il flusso del procedimento dal suo avvio (Istanza, dichiarazione, segnalazione, avvio d'ufficio) fino al suo completamento, evidenziando eventuali criticità
	Stati previsti	Una volta avviato il procedimento questo potrebbe assumere degli stati (per esempio preso in carico, rifiutato, accettato)
	Durata del procedimento	Indicare i tempi del procedimento amministrativo individuati anche sul portale trasparenza
	Settori, servizi o uffici coinvolti nel procedimento	Indicare quali altri settori, servizi o uffici sono coinvolti per il completamento del procedimento
	Organizzazioni ed enti esterni coinvolti nel procedimento	Indicare quali organizzazioni o enti esterni sono coinvolti per il completamento del procedimento
	Software utilizzato per la gestione del procedimento	Indicare l'eventuale piattaforma o software per la gestione del procedimento

Terminata questa prima parte dell'attività sarà possibile procedere all'analisi dei documenti prodotti all'interno di ogni procedimento amministrativo.



La seguente proposta di modulo andrà compilata per ogni documento collegato al procedimento; che si tratti di un'istanza, di un documento allegato all'istanza, o un documento prodotto/ricevuto nell'espletamento del procedimento questo sarà descritto in tutte le sue parti.

Oggetto dell'analisi	Elemento	Descrizione
Documento	Oggetto del documento	Indicare sinteticamente il nome del documento (ad es. Richiesta di accesso agli atti, Concessione di occupazione del suolo pubblico, etc)
	Tipologia di documento	Indicare la tipologia di documento (ad esempio: Istanza di parte, comunicazione, concessione, autorizzazione, segnalazione, dichiarazione, parere, convenzione, contratto, altro...)
	Modulo o modello esistente	Indicare se il documento viene prodotto a partire da un modulo (modulistica pdf, modulo online) o se a partire da un modello
	Quantità di documenti per anno	Indicare il numero di documenti prodotti annualmente
	Supporto e formato del documento	Indicare se il documento viene prodotto su supporto cartaceo o se digitale indicando il formato (pdf, p7m, docx, odt, xls, etc...)
	Strumento utilizzato per la produzione del documento	Scegliere dall'elenco degli applicativi in uso dall'organizzazione quello utilizzato per produrre il documento
	Tipologia sottoscrizione del documento	- Firma autografa - Firma digitale (cades, pades, xades)
	Produttore del documento	Scegliere tra 1 – Il mio ufficio 1 - Organigramma dell'organizzazione 2 – Azienda, cittadino, ente pubblico esterno
	Destinatario del documento	Indicare il destinatario del documento
	Altri supporti utilizzati per la gestione del documento	Indicare se una volta prodotto o ricevuto, il documento viene stampato, scansionato per la sua gestione (smistamento, allegato al protocollo, fascicolazione)
	Registrazione di protocollo	Indicare se il documento è soggetto a registrazione di protocollo (scegliere tra NO, Entrata, Uscita, Interno)
	Codifica del protocollo	Indicare se all'interno di p@doc esiste una codifica
	Categorie particolari di dati personali	Indicare se il documento contiene particolari categorie

	(Ex. Art. 9 e 10)	di dati personali
	Classificazione archivistica	Indicare la classificazione archivistica (piano di classificazione)
	Ulteriori registrazioni in repertori o registri particolari	Indicare se il documento è soggetto a registrazione particolare (repertori o registri)
	Fascicolo (tipologia)	Indicare se il documento viene inserito all'interno di un fascicolo e se si la tipologia di fascicolo - no - per affare - per attività - per persona fisica o giuridica
	Fascicolo (classificazione)	Indicare la classificazione archivistica del fascicolo
	Fascicolo (accessibilità e ACL)	Indicare quali settori-uor-uffici possono accedere o popolare il fascicolo
	Obblighi di pubblicazione	Indicare se il documento è soggetto ad obblighi di pubblicazione (albo, trasparenza, altro..)
	Accessibilità al documento	Indicare quali soggetti (coinvolti o estranei al procedimento) possono accedere al documento
	Conservazione del documento	Indicare: - Tempo di conservazione - Tempo disponibilità del documento nel sistema di gestione

Terminata l'attività sarà possibile quantificare la propria produzione documentale, individuare quali documenti sono prodotti su supporto digitale e quanti su supporto analogico, analizzare la conformità dei documenti prodotti rispetto le linee guida e rispetto alle politiche di gestione documentale definite dall'organizzazione, nonché identificare i sistemi di produzione documentale ed il grado di copertura del protocollo informatico.

L'attività di riconoscimento consentirà agli utenti di acquisire una maggiore consapevolezza sulla propria attività e di attivare dei processi di semplificazione e di miglioramento del lavoro grazie alla revisione dei processi amministrativi e al consolidamento delle buone pratiche.

3.2 IL PIANO DI CLASSIFICAZIONE

Dopo la riconoscimento precedentemente proposta sarà possibile procedere con la revisione o la validazione del proprio piano di classificazione (titolario). Lo standard ISO 15489-1:2016 *Information and documentation – Records*

management definisce le funzioni e i processi ai quali le informazioni (ovvero i documenti) devono essere sottoposti per poter rispondere ai requisiti di autenticità, integrità, affidabilità e usabilità.

Una di queste funzioni è proprio la classificazione, ossia quell'operazione finalizzata a collegare il documento al suo contesto (funzione, attività o processo) e di determinarne la corretta fascicolazione e sedimentazione all'interno dell'archivio digitale. Il legislatore stesso la definisce come un'operazione necessaria per la tenuta del sistema di gestione documentale (art. 56 del DPR 445/2000) attraverso il quale possiamo definire a monte le politiche di conservazione dei documenti.

All'attività di ricognizione dei procedimenti amministrativi e dei documenti si dovrà necessariamente integrare quindi un'attività di redazione, revisione o semplice validazione (se già in uso) di un titolario di classificazione aggiornato e che rispecchi le funzioni dell'organizzazione. Quest'ultimo, oltre ad assolvere alla sua funzione di corretta sedimentazione e aggregazione dei documenti nell'archivio, è il presupposto per l'utilizzo del piano di conservazione, ovvero dello strumento che definisce in base alla norma i tempi di conservazione delle tipologie documentarie prodotte dal soggetto produttore.

3.3 ASPETTI DELLA CONSERVAZIONE NELLA FORMAZIONE DEL DOCUMENTO INFORMATICO E DEL FASCICOLO INFORMATICO

Le attività precedentemente descritte produrranno dei dati essenziali e indispensabili per implementare un set di metadati completo e funzionale all'invio in conservazione. La funzione dei metadati, così come definita anche dallo standard ISO 23081 – *Metadata for records*, non è esclusivamente quella di identificare il documento nella sua prima registrazione ma anche di governarne l'intero ciclo di vita definendo una vera e propria stratificazione informativa. Nello specifico possiamo identificare le seguenti tipologie di metadati:

- a) metadati sul documento stesso (identificazione, registrazione, accessibilità e sicurezza)
- b) metadati sulle norme, regole, policy e disposizioni (conformità alle norme, alle politiche interne e alla classificazione adottata)
- c) metadati sulle persone (persone coinvolte nella creazione e registrazione del documento, ruoli e autorizzazioni di accesso nell'intero processo)
- d) metadati sull'attività e sui processi (metadati sulle funzioni, le attività e le transizioni nei processi e nei procedimenti)
- e) metadati sul processo di gestione documentale (metadati per automatizzare i processi, le aggregazioni e i metadati relativi alla conservazione)



In questo contesto, il sistema di protocollo informatico e di gestione documentale [p@doc](#) si avvale dello strumento “oggettario” attraverso il quale è possibile associare al documento e all’aggregazione documentale un set di metadati completo ed esaustivo. Alle operazioni di creazione dei record attraverso le quali vengono registrate le informazioni minime previste dalla norma, possiamo aggiungere ulteriori metadati che agevolano la corretta gestione e archiviazione del documento.

Di seguito proponiamo un modello base della struttura dell’oggettario dei documenti :



NUOVA VERSIONE OGGETTARIO DEI DOCUMENTI

TABELLA OGGETTARIO	DESCRIZIONE
pgo_Id	Id dell'oggetto
Pgo_CodAmm	Indica il codice dell'amministrazione
Pgo_Descrizione	Descrizione dell'oggetto
pgo_Stato	Stato dell'oggetto – Attivo o disattivo
Pgo_k_titolario	Riferimento ID titolario
pgo_k_assegnazione	Riferimento UO di smistamento
pgo_categoria	Fa riferimento all'albo
pgo_publishable	Indica se il protocollo è da pubblicare o meno all'albo
pgo_publication_days	Giorni di pubblicazione all'albo
id_Registro	Indica il nome o il codice del registro (Ad esempio Registro di Protocollo, Repertorio dei contratti, Registro delle fatture etc..)
id_Processo	Indica il codice del procedimento amministrativo all'interno dell'organizzazione
id_Fascicolo	Indica il codice dell' oggettario fascicoli
Protezione_dati	Indica il livello di riservatezza dei dati personali o dei dati di interesse privato
Riservatezza	Indica il livello di riservatezza dei dati dell'organizzazione o dei dati di interesse pubblico
Tempo_Conservazione	Indica il numero di anni in cui il documento deve essere conservato
Tipologia_Documentaria	Indica la tipologia documentaria al quale appartiene il documento registrato in relazione alle specifiche definite dal piano di conservazione
Tipologia_SIP	Indica la tipologia di pacchetto di versamento definito da produttore e conservatore al quale fa riferimento il documento registrato



Allo stesso modo, anche i fascicoli e le aggregazioni documentali devono essere create associando loro dei metadati minimi prevista dalla norma ed ulteriori metadati utili a governare l'intero ciclo dell'aggregazione e dei documenti in essa contenuti. Di seguito il modello base di oggettario dei fascicoli e delle aggregazioni documentali:

NUOVA VERSIONE OGGETTARIO FASCICOLI	
TABELLA OGGETTARIO	DESCRIZIONE
Id	Id dell'oggetto fascicolo
Descrizione	Descrizione dell'oggetto fascicolo
Titolario	Riferimento ID titolario
Visibilità	Predetermina la visibilità del fascicolo alla/alle UOR competenti
Tipologia_Fascicolo	Indica la tipologia di fascicolo (Entità, Affare, attività)
id_Processo	Indica il codice del procedimento amministrativo all'interno dell'organizzazione
Protezione_dati	Indica il livello di riservatezza dei dati personali o dei dati di interesse privato
Riservatezza	Indica il livello di riservatezza dei dati dell'organizzazione o dei dati di interesse pubblico
Tempo_Conservazione	Indica il numero di anni in cui il documento deve essere conservato
Tipologia_SIP	Indica la tipologia di pacchetto di versamento definito da produttore e conservatore al quale fa riferimento il fascicolo o l'aggregazione documentale.



4. LA CONSERVAZIONE NEGLI ENTI DI PRODIGIO

L'implementazione della componente dell'archivio storico all'interno del sistema di gestione documentale e di protocollo informatico [p@doc](#) deve considerare dapprima dell'aspetto organizzativo tenendo conto soprattutto delle prescrizioni contenute nella norma e sulle tipologie documentali definite per l'invio in conservazione in base ai servizi di conservazione adottati.

Il modello organizzativo adottato dai partner di progetto è quello definito dall'art. 5 , comma 2 b) del Decreto del Presidente del Consiglio dei Ministri 3 dicembre 2013 – *Regole tecniche in materia di sistema di conservazione* ovvero un modello di conservazione in outsourcing dove il servizio di conservazione viene affidato ad altri soggetti, pubblici o privati che offrono idonee garanzie organizzative e tecnologiche e nel caso specifico che siano accreditati presso l'Agenzia per l'Italia digitale.

In particolare i servizi di conservazione adottati dal partenariato sono i seguenti:

- **PARER** (Polo Archivistico Regione Emilia Romagna)
- **DOCFLY** (Aruba PEC)
- **Conservazione No Problem** (Arancia ICT)
- **LegalDoc** (Inforcert)

Le tipologie documentali che verranno confrontate ed analizzate in un primo momento saranno:

- Registro giornaliero di protocollo
- Fattura elettronica Attiva
- Fattura elettronica passiva



5. TAKO- ANALISI DEL MODELLO APPLICATIVO PER IL VERSAMENTO DEI DOCUMENTI

5.1 Analisi del contesto

TAKO (Trasmissione Archivi in Conservazione) è uno dei progetti del Comune di Padova che rientrano nell'attività di progettazione e sviluppo della nuova piattaforma per la gestione documentale dell'ente.

Nella sua prima versione TAKO è stato realizzato come modulo integrato (e non indipendente) all'interno del sistema di protocollo e gestione documentale “P@DOC” del Comune di Padova. Nelle fasi future è prevista la sua re-ingegnerizzazione e re-implementazione basandosi sulla nuova infrastruttura che verrà realizzata nell'ambito del nuovo sistema di gestione documentale. TAKO diverrà così una applicazione autonoma ed integrata con gli altri moduli previsti dalla futura infrastruttura (p@org – organigramma, documentale, etc.).

Gli obiettivi principali del progetto sono:

- invio di documenti informatici prodotti mediante i vari software dell'ente ad uno o più soggetti conservatori accreditati (Parer, Aruba, Infocert, Arancia ICT)
- corretta metadatazione e classificazione dei documenti al fine di una corretta conservazione (garantire certezza nella ricerca dei documenti in futuro)
- realizzare l'infrastruttura che permetta la corretta gestione dell'archivio storico informatico dell'ente (ricercare documenti in conservazione, reperire i documenti conservati)

La realizzazione di tale software permetterà di:

- Creare un'unica infrastruttura per la gestione dell'archivio storico informatico dell'ente
- Definire un unico standard mediante il quale i vari software “produttori” possano inviare in conservazione i documenti indipendentemente dal soggetto conservatore a valle
- Fornire strumenti per la corretta classificazione, metadatazione, definizione dei parametri di conservazione ed invio al conservatore dei documenti
- Realizzare strumenti di supporto per l'invio in conservazione di documentazione non gestita per mezzo di software verticali (documenti in share/cartelle condivise)
- Strumenti di migrazione dei documenti conservati (spostare da un conservatore ad un altro)

Al momento della stesura del presente documento di pre-analisi la normativa prevede la standardizzazione del solo pacchetto di archiviazione secondo lo standard UNI 11386:2010 Standard SInCRO - *Supporto all'Interoperabilità nella Conservazione e nel Recupero degli Oggetti digitali*, ossia del pacchetto che gli enti conservatori accreditati sono tenuti a conservare correttamente presso i propri sistemi informatici.



Non è invece prevista la definizione di un unico standard, sia in termini di metadati che di colloquio informatico fra i sistemi, nelle varie fasi di:

- invio dei documenti al conservatore
- verifica dello stato di conservazione
- esibizione/recupero di un documento già posto in conservazione

Nell'ambito del progetto è stata condotta l'analisi ed il confronto delle specifiche previste da ciascuno dei soggetti conservatori adottati dal partenariato e nello specifico:

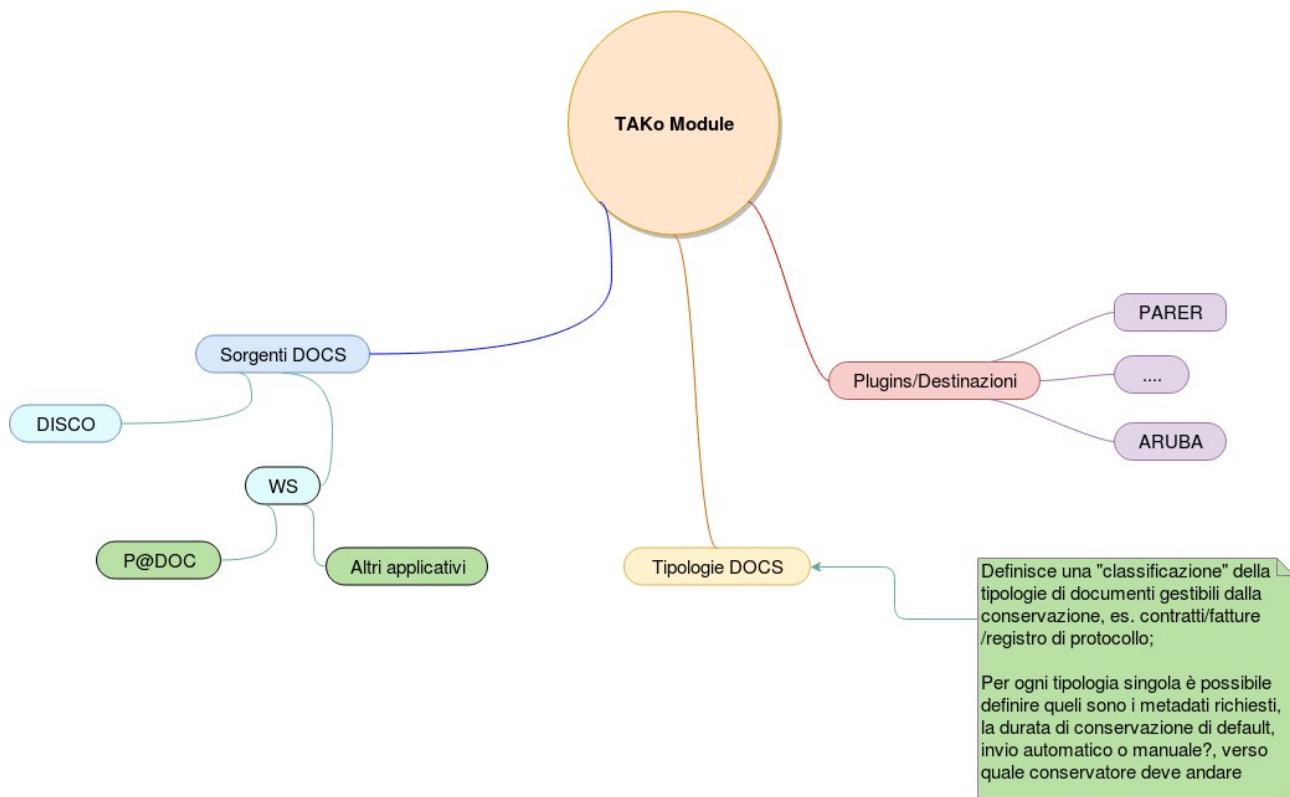
- Confronto tra le specifiche WS e la macrostruttura dei pacchetti di versamento (allegato 1 - confronto_conservatori_ws_e_pdv.ods)
- Confronto metadati fattura elettronica passiva (Allegato 2 - METADATAZIONE TAKO - FATTURA PASSIVA.ods)

Al termine di tale analisi il modello dati previsto da Parer è risultato essere il più completo in quanto consente di descrivere correttamente la documentazione prodotta dalla pubblica amministrazione e risulta allineato con le specifiche di progetto sia dell'attuale sistema di gestione documentale P@DOC che del futuro sistema di gestione documentale.

Parer oltre ad essere un conservatore accreditato si distingue per essere un soggetto pubblico (IBACN – Istituto Beni Artistici Culturali Naturali Emilia-Romagna), il quale svolge il servizio di conservazione per un elevato numero di soggetti pubblici. Per questo motivo il suo standard può considerarsi uno standard de facto.

5.2 Concept di massima

L'immagine che segue rappresenta un concept di massima di TAKo



TAKo si pone come intermediario tra i vari applicativi dell'ente ed i vari enti conservatori con i quali il Comune ha sottoscritto dei contratti specifici per la conservazione a norma.

TAKo permetterà di ricevere dai vari software produttori i documenti che devono essere inviati in conservazione esponendo uno o più canali di INPUT, esempio:

1. WebServices REST esposte da TAKo
2. Cartella condivisa

In merito alle WebServices si prevede la realizzazione di una serie di chiamate HTTP che permettano ai vari software produttori di integrarsi correttamente con TAKo. Le specifiche di tale WS saranno oggetto dell'attività di progettazione che verrà realizzata nel proseguo del progetto TAKo.

I metadati scambiati mediante tale WS dovranno essere conformi agli XML previsti da Parer ("Parer nativo"). E' comunque possibile che, in seguito all'attività di analisi, si renda necessario estendere con nuovi metadati e strutture logiche le specifiche attualmente da Parer al fine di garantire particolare funzionalità di TAKo ("Parer esteso"). In questa fase di pre-analisi non è possibile stabilire a priori questi aspetti.



Nel seguito del presente documento sono da intendersi “metadati in formato TAKo” i metadati scambiati fra TAKo e i software produttori nel rispetto dello standard “Parer nativo” o “Parer esteso” (vedi quanto espresso nel precedente capoverso).

Una adeguata classificazione dei documenti mediante le “Tipologie DOCS” permetterà di garantire la corretta metadatazione degli stessi ed il rispetto dei tempi di conservazione previsti dalle norme in vigore.

TAKo permetterà inoltre, nel caso in cui siano disponibili più enti conservatori, definire per distinte tipologie documentali, distinti soggetti conservatori. Quest’ultimo aspetto potrebbe garantire risparmi economici all’ente in funzione delle tariffe previste dai vari soggetti conservatori.

La comunicazione fra i sistemi informatici dei singoli soggetti di conservazione è interamente gestita da TAKo il quale dovrà:

- Recepire il pacchetto da versare in conservazione, (files e metadati in formato Tako), dal software produttore verificandone la correttezza formale
- Riconoscere la “tipologia documentale” di appartenenza del pacchetto, individuando il conservatore previsto per essa e i parametri di conservazione (tempi)
- Effettuare una conversione dal formato dei metadati (ed il modello dati) di Tako al formato dei metadati (e modello dati) previsto dal soggetto conservatore
- Gestire le comunicazione con il sistema informatico del soggetto conservatore secondo le specifiche previste da quest’ultimo garantendo una corretta gestione degli stati di errore
- Fornire lo stato/esito dell’invio in conservazione del pacchetto al software produttore

Sono da prevedere in TAKo specifiche interfacce grafiche per:

- Configurare TAKo
- Monitorare lo stato dei vari pacchetti in ingresso a TAKo
- Monitorare lo stato degli invii in conservazione
- Ricercare documenti già in conservazione
- Recuperare/esibire documenti già in conservazione
- tool di migrazione fra soggetti conservatori diversi (di intere tipologie documentarie?)



5.3 Macro struttura SIP Parer

Come indicato in precedenza si è stabilito di considerare come “standard de facto” le specifiche previste da Parer. TAKo riceverà pertanto i documenti da inviare in conservazione utilizzando una struttura dati e metadati compatibile con quella prevista da Parer per i loro sistemi.

L’analisi per la progettazione e realizzazione di TAKo si basa sulle specifiche di Parer versione 2.3 (23/5/2018). Per maggiori dettagli sulle specifiche di Parer si prega di fare riferimento alla relativa documentazione (“specifiche servizi di versamento v2.3”).

Il presente paragrafo si prefigge di individuare le principali caratteristiche previste dal modello dati e dei metadati definite da Parer e pertanto “ereditate” da TAKo.

5.3.1 SIP/Unità documentaria

Con il termine SIP (Submission Information Package) si identifica il pacchetto di versamento che deve essere inoltrato al sistema informatico di Parer per effettuare l’invio in conservazione di un insieme di documenti.

Un pacchetto SIP coincide con una singola unità documentaria, l’unità documentaria è la struttura minima che permette la corretta descrizione di un documento (es. una fattura, una determina, un generico documento protocollato).

Per ogni unità documentaria che si intende inviare in conservazione dovrà essere prodotto il relativo pacchetto SIP. Ciascun pacchetto SIP verrà inviato a TAKo secondo le specifiche WS che verranno implementate. I metadati presenti nel SIP dovranno essere codificati per mezzo di un apposito file XML (XMLSIP) nel rispetto dello standard “metadati TAKo”.

L’XML SIP permette di descrivere l’intera struttura dell’unità documentaria (unità documentaria, documenti, componenti e sotto-componenti) e i relativi metadati.

La seguente immagine schematizza la struttura di un SIP secondo quanto previsto da Parer:

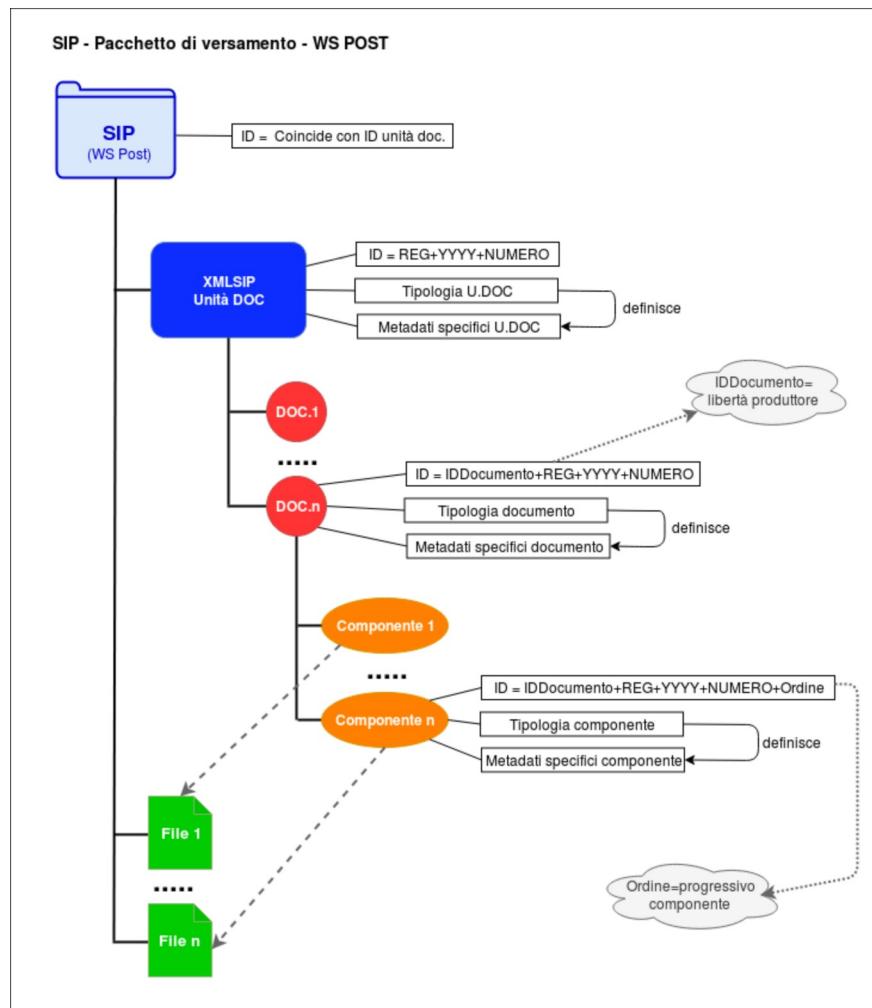


Figure 1

- Un file “XML SIP”, l’insieme dei metadati/informazioni che permettono di rappresentare l’unità documentaria, (e ne permetteranno la ricerca in futuro), vengono codificati in un file XML secondo lo “standard TAKo”
- L’insieme dei file (documenti), oggetto della conservazione, che appartengono l’unità documentaria

Ogni pacchetto SIP viene identificato con un codice univoco, tale codice coincide con l’ID assegnato all’unità documentaria.

L’unità documentaria è identificata mediante un codice univoco (ID) assegnato dal software produttore, tale ID sarà composto da:



1. Codice registro, un codice identificante il registro di appartenenza dell'unità documentaria (es. registro di protocollo)
2. Anno, l'anno nel quale l'unità documentaria è stata registrata nel precedente registro
3. Numero, il numero progressivo univoco assegnato all'unità documentaria all'interno del suddetto registro

L'unità documentaria viene “classificata” e “tipizzata” assegnandole una “tipologia di unità documentaria”, es:

- Fattura attiva
- Fattura passiva
-
- Determina
- Delibera
- Generico documento protocollato

Lo standard dei metadati prevede, per l'unità documentaria, due set di informazioni:

1. metadati “fissi”, stabiliti a livello dello “standard TAKo”, invariabili al variare della tipologia documentaria assegnata
2. metadati “specifici”, variabili in quanto legati alla specifica tipologia di unità documentaria; distinte tipologie documentarie prevedono distinti metadati specifici

5.3.2 Documenti

Ogni unità documentaria raggruppa più elementi logicamente appartenenti all'aggregazione rappresentata dalla stessa unità, ciascun elemento è un documento archivistico. Per documento archivistico si intende l'insieme di informazioni memorizzate su un qualsiasi supporto, prodotte o ricevute e conservate da un ente o da una persona nello svolgimento delle proprie attività o nella condotta dei propri affari.

Ogni documento è identificato mediante un “IDDocumento”, un identificativo che deve essere univoco (almeno) a livello di unità documentaria, tale identificativo viene assegnato dal software produttore dell'unità documentaria.



I documenti, all'interno dell'unità documentaria, vengono classificati prima di tutto in funzione delle seguenti "macro categorie" (elementi principali dell'unità documentaria):

- Documento "principale" (obbligatorio, uno ed uno solo documento), definisce il contenuto primario dell'unità documentaria
- Allegato (nessuno o più documenti), integra le informazioni contenute nel documento principale. E' redatto contestualmente o precedentemente al documento principale
- Annesso (nessuno o più documenti), fornisce ulteriori informazioni o notizie a corredo del documento principale, generalmente prodotto e inserito in un momento successivo a quello di creazione dell'unità documentaria
- Annotazione (nessuna o più documenti), riporta elementi identificativi dell'unità documentaria e del suo iter documentale (es. segnatura di protocollo)

Tali "macro categorie" sono disponibili per qualsiasi unità documentaria, indipendentemente dalla sua tipologia, esse sono stabilite a livello di "standard TAKo" (parer).

Ciascun singolo documento (principale, allegato, annesso o annotazione) viene "classificato" e "tipizzato", definendone la natura mediante l'assegnazione di una "tipologia documento", es:

- Fattura
- Determina
- Parcella
- Notifica esito committente
- Relata pubblicazione
- Visto regolarità contabile
- Ricevuta di consegna

La tipologia documentaria, dato che determina la natura dell'unità documentaria, stabilisce anche la natura dei documenti che vi possono far parte. La tipologia documentaria stabilisce perciò quali tipologie di documenti che possono essere utilizzate per classificare i documenti che ne fanno parte.

Ad esempio, per la tipologia documentaria "Fattura passiva", saranno disponibili le seguenti tipologie documentarie:

- Fattura
- Parcella
- Notifica esito committente



Non saranno invece utilizzabili:

- Determina
- Relata pubblicazione
- Visto regolarità contabile
- Ricevuta di consegna

In funzione della tipologia documentaria viene inoltre stabilito se una determinata tipologia di documento possa essere usata per classificare:

- il documento principale
- oppure uno (o più) degli allegati
- oppure uno (o più) degli annessi
- oppure una (o più) delle annotazioni

In sintesi, all'interno di una determinata tipologia documentaria, una specifica tipologia di documento, può essere associata solo ad una ed una sola delle precedenti macro categorie di documenti.

Consideriamo ad esempio la tipologia documentaria “fattura passiva”, il seguente schema rende espone le tipologie documentarie disponibili in funzione di ciascuna “macro categoria”:

Elemento/Macro categoria	Tipologie di documento disponibili
Documento principale	<ul style="list-style-type: none">• Fattura• Acconto/Anticipo su fattura• Acconto/Anticipo su parcella• Nota di credito• Nota di debito• Parcella
Allegati	<ul style="list-style-type: none">• Generico
Annesso	<ul style="list-style-type: none">• Transcodifica fattura• File dei metadati• Notifica esito committente• Scarto esito committente• Notifica decorrenza termini• Integrazione reverse charge

Come evidenziato dal precedente prospetto, un documento può essere classificato con la tipologia di documento “fattura” se e solo se il documento in oggetto è il “documento principale” dell’unità

documentaria. Allo stesso modo un documento può essere classificato con la tipologia di documento “file dei metadati” se e solo se il documento analizzato è uno dei possibili “annessi” dell’unità documentaria.

Lo standard dei metadati prevede, per ogni documento, due set di informazioni:

1. metadati “fissi”, stabiliti a livello dello “standard TAKo”, invariabili al variare della tipologia di documento
2. metadati “specifici”, variabili in quanto legati alla specifica tipologia di documento; distinte tipologie di documento prevedono distinti set di metadati specifici

5.3.3 Componente

Ciascun documento appartenente all’unità documentaria, (indipendentemente che sia il documento principale, uno degli allegati, uno degli annessi o delle annotazione), può essere articolato in uno o più componenti, ciascun componente coincide sostanzialmente con un file necessario alla sua rappresentazione informatica.

Ad ogni componente viene assegnato un identificativo progressivo univoco “OrdinePresentazione” tale valore deve essere univoco a livello di documento. Viene inoltre associato un identificativo denominato “ID” che deve invece essere univoco a livello di SIP/Unità documentaria, quest’ultimo ID coincide sostanzialmente col nome del file al quale il componente fa riferimento.

Prendiamo in considerazione una unità documentale “fattura passiva”, in particolare il documento principale costituito dalla fattura ricevuta, tale documento può essere costituito da:

Componente	Tipo supporto	Tipo contenuto	Formato file	Descrizione
C1	File	Contenuto	XML	XML della fattura elettronica ricevuta
C2	File	Rappresentazione	PDF	Rappresentazione in PDF della medesima fattura

Sono previste le seguenti tipologie di componenti:

- Contenuto, il componente fa riferimento al file che costituisce il documento
- Firma, indica che il componente si riferisce al file contenente la firma (separata/detached) del documento (contenuto)



- Marca, indica che il componente si riferisce al file contenente la marca temporale (separata/detached) del documento (contenuto)
- Rappresentazione, indica che il componente si riferisce al file contenente una rappresentazione del documento (contenuto)

Ciascuna componente di un documento viene classificata mediante una delle voci di “tipologia componente” sopra riportate al fine di stabilire la natura del file in relazione al documento.

Lo standard dei metadati prevede, per ogni componente, due set di informazioni:

1. metadati “fissi”, stabiliti a livello dello “standard TAKo”, invariabili al variare della tipologia di componente
2. metadati “specifici”, variabili in quanto legati alla specifica tipologia di componente; distinte tipologie di componente prevedono distinti set di metadati specifici

In funzione della natura del documento (tipologia documento) viene stabilito quali tipologie di componenti possano costituire il documento, tenendo presente che la componente “contenuto” deve essere sempre presente (è ciò che costituisce il documento).

5.3.4 Sotto-componente

Ogni singola componente, di un determinato documento dell’unità documentaria, può essere costituita a sua volta da più sotto-componenti.

Prendiamo in considerazione la componente “XML della fattura elettronica” del precedente esempio, potrebbe presentarsi la seguente situazione:

Componente/Sottocomponente	Tipo supporto	Tipo contenuto	Formato file	Descrizione
C1	File	Contenuto	XML	XML della fattura elettronica ricevuta
C1.1 (sottocomponente)	File	Foglio di trasformazione	XSLT	Foglio di trasformazione della fattura

In questo modo:

- Il file XML rappresenta il vero contenuto del documento, essendo la fattura elettronica vera e propria
- Il file XSLT, che contiene istruzioni informatiche per la conversione dell’XML in una versione HTML leggibile dagli umani, rappresenta una sotto-componente (secondaria) del fattura stessa

Per le sotto-componenti valgono le medesime considerazioni in precedenza, inoltre è da tenere conto che, la tipologia di “componente” stabilisce quali tipologie di sotto-componenti possano essere presenti.

La sotto-componente “Foglio di trasformazione della fattura” è prevista solo per la componente “fattura”, ma essa non è obbligatoria.

5.3.5 Sintesi struttura

Possiamo quindi sintetizzare quanto precedentemente indicato:

1. Pacchetto SIP:

- A. il pacchetto SIP coincide con una ed una sola unità documentaria
- B. la struttura dell'unità documentaria (e i relativi metadati) viene rappresentata mediante un file XML inserito nel SIP (XMLSIP)

2. Unità documentaria:

- A. ciascuna unità documentaria è identificata con un ID univoco (lato produttore) costituito da:
 - a. registro
 - b. anno registrazione
 - c. progressivo registrazione
- B. l'unità documentaria possiede dei metadati "fissi" previsti dallo standard
- C. l'unità documentaria viene classificata per mezzo di una "tipologia unità documentaria" che stabilisce:
 - a. quali siano i metadati specifici per quella unità documentaria
 - b. quali tipologie di documento possano essere presenti nell'unità documentaria e per quale elemento (documento principale, annesso,) la data tipologia di documento possa essere usata
- D. l'unità documentaria è costituita da uno o più documenti (elementi):
 - a. un documento principale (obbligatorio)
 - b. zero o più allegati
 - c. zero o più annessi
 - d. zero o più annotazioni

3. Documento:

- A. ciascun documento è identificato mediante un "IDDocumento", valore univoco (almeno) a livello di unità documentaria, è assegnato dal software produttore
- B. il documento possiede dei metadati "fissi" previsti dallo standard
- C. il documento viene classificato per mezzo di una "tipologia documento" che stabilisce:
 - a. quali siano i metadati specifici per quella tipologia di documento
 - b. quali componenti e relativi sotto-componenti (di ciascun componente) possano costituire il documento
- D. ogni documento può essere composto da uno o più componenti, ogni componente identifica un file (relativo al documento) che deve essere inviato in conservazione
- E. ogni documento deve avere almeno un componente di tipo (tipo componente) "contenuto"

4. Componente:

- A. ogni componente possiede un identificativo progressivo "OrdinePresentazione" univoco a livello di singolo documento. Vi è inoltre l'identificativo "ID" (univoco a livello di SIP/Unità documentaria) che coincide col nome del file
- B. il componente possiede dei metadati "fissi" previsti dallo standard
- C. il componente viene classificato per mezzo di una "tipologia componente" che stabilisce:
 - a. quali siano i metadati specifici per quella tipologia di componente



- D. le principali “tipologie componente” sono:
- contenuto (il file che costituisce il documento), componente obbligatorio per ogni documento
 - firma, firma (separata/detached) del documento (contenuto), componente facoltativo per ogni documento
 - marca, marca temporale (separata/detached) del documento (contenuto), componente facoltativo per ogni documento
 - rappresentazione, una rappresentazione del documento (contenuto), componente facoltativo per ogni documento
- E. Ogni componente può essere costituito da uno o più sotto-componenti, in funzione della tipologia di documento
5. Sotto-componente:
- ogni sotto-componente possiede un identificativo progressivo “OrdinePresentazione” univoco a livello di singolo componente. Vi è inoltre l’identificativo “ID” (univoco a livello di SIP/Unità documentaria) che coincide col nome del file
 - Valgono le medesime considerazioni fatte per il componente, non è però obbligatorio che vi sia almeno un sotto-componente di tipo “contenuto” (per ogni “componente” di appartenenza)
 - Non sono previsti ulteriori sotto-livelli (sotto-sotto-componenti non sono previsti)



5.4 Schema e/r di massima

Lo schema e/r che segue rappresenta schematicamente quanto precedentemente illustrato per l'unità documentaria/SIP. Si tratta di uno schema di massima la cui finalità è rappresentare le principali caratteristiche e relazioni esistenti fra le varie entità che costituiscono il modello dati dell'unità documentaria/SIP.

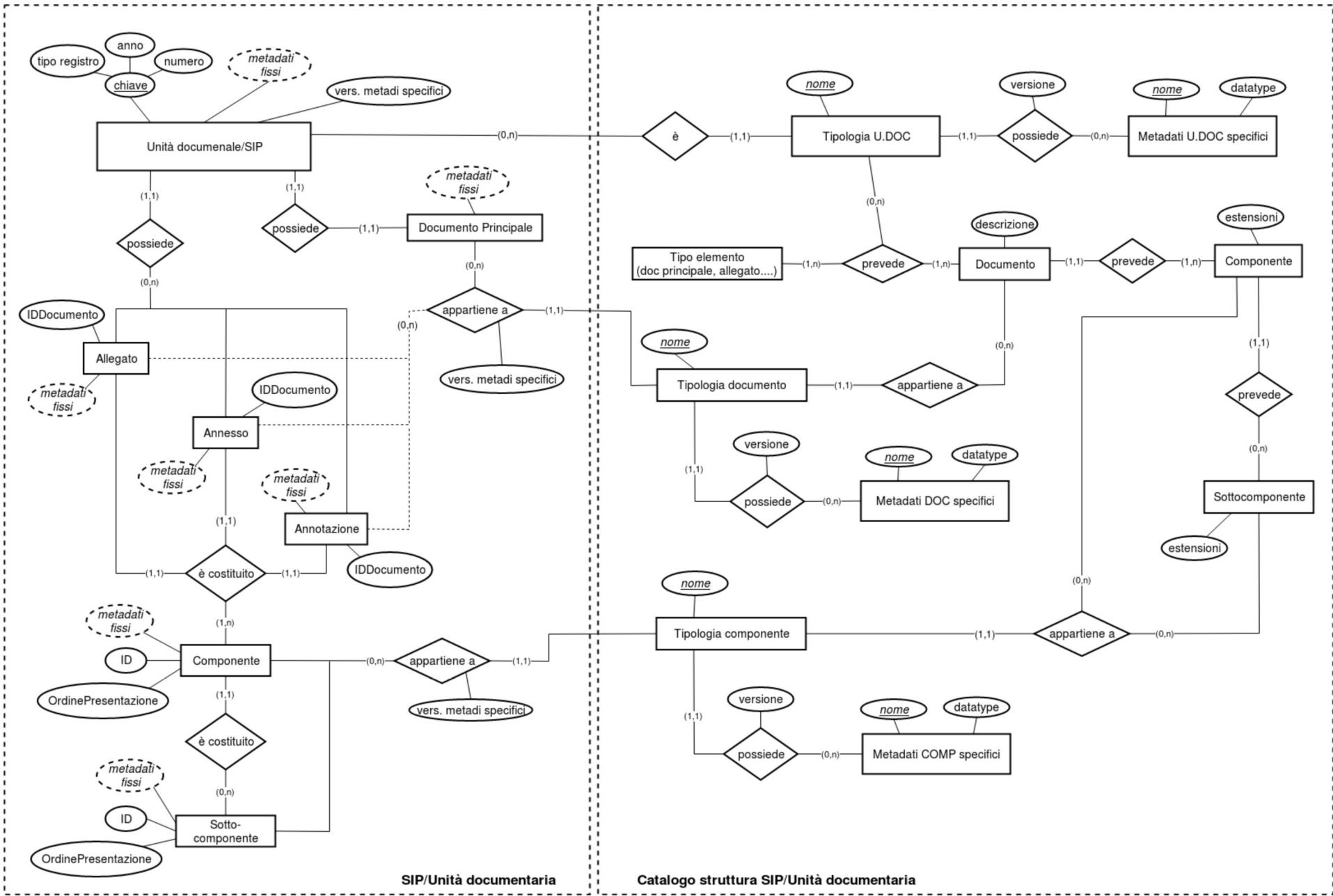


Figure 2

5.5 Flusso di lavoro, concept del processo

TAKo, nella sua versione finale, si pone come intermediario tra un software produttore di documenti e un sistema di conservazione a norma.

In Italia esistono diversi conservatori a norma, ciascun conservatore possiede un proprio sistema informatico per la conservazione dei documenti ed ognuno di questi presenta differenti specifiche tecniche per le operazioni di versamento o esibizione dei pacchetti di archiviazione.

TAKo dovrà perciò sviluppare, per ciascuno di questi enti di conservazione, specifici connettori che implementino i protocolli di comunicazione previsti dalle singole specifiche tecniche. Questo permetterà di supportare diversi conservatori e di poter impostare, in funzione della tipologia documentaria, verso quale conservatore inviare una data tipologia di documenti.

TAKo dovrà sostanzialmente “convertire” il pacchetto SIP (in formato TAKo), inviatogli da un determinato software produttore dell’ente, nel corrispettivo pacchetto di versamento (PdV) previsto dall’ente conservatore prescelto. Al termine di questa “conversione” TAKo gestirà l’iter di versamento del PdV all’ente conservatore (con relativa gestione degli esiti ed eventuali errori). Considerando che le specifiche (formato PdV e il protocollo comunicazione-WS) per l’invio in conservazione variano da conservatore a conservatore, TAKo devolverà le operazioni di conversione (dal SIP al PdV) e di gestione della comunicazione (col sistema di conservazione) allo specifico connettore/plugin.

Il processo di “parsing” (lettura e validazione del pacchetto SIP) rimarrà invece in capo al nucleo di TAKo.

Il presente paragrafo illustra il “concept”, il concetto di “base”, individuato per il processo di elaborazione dei dati. La successiva fase di progettazione, che porterà a dettagliare il processo di “parsing-conversion” (lettura/validazione SIP dati in ingresso - conversione in PdV d’uscita), si baserà sostanzialmente su quanto riportato nel seguito.

Il flusso di lavoro di TAKo può sostanzialmente essere riassunto in queste macro fasi:

1. Ricezione del pacchetto SIP dal produttore
 - A. Parsing
 - validazione metadati
 - estrazione dei metadati
 - rappresentazione metadati in una struttura dati elaborabile agevolmente da TAKo (Object informatico)
 - B. Memorizzazione/indicizzazione sui sistemi TAKo
2. Invio al conservatore
 - A. Caricamento dati memorizzati in TAKo, riottenimento metadati in una struttura dati elaborabile agevolmente da TAKo (Object informatico)
 - B. Individuazione del conservatore per la tipologia documentale di appartenenza del SIP, caricamento del corrispondente connettore/plugin
 - C. Produzione del pacchetto di versamento previsto (conversion)
 - D. Invio al conservatore

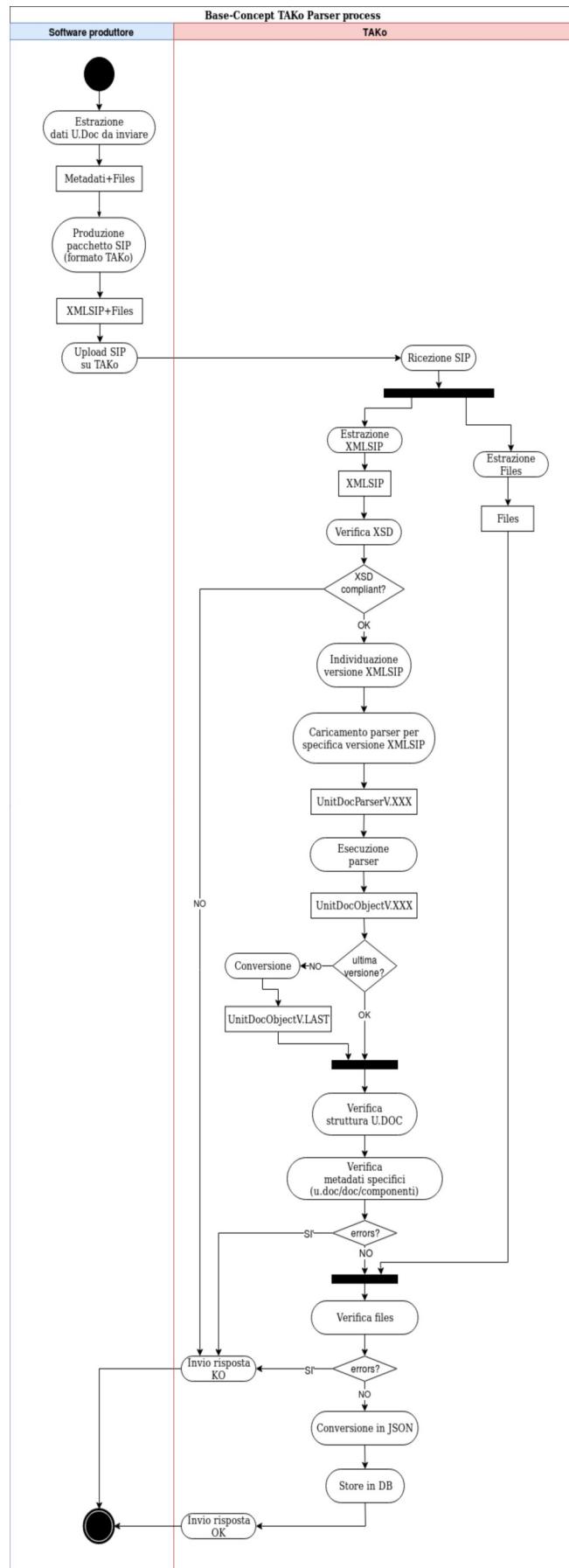
E. Gestione stato invio

Le due fasi “Ricezione SIP” ed “Invio in conservazione” verranno implementate su processi distinti ed isolati che verranno eseguiti in tempi differenti; questa scelta permette di garantire integrità nei dati e “chiudere” il prima possibile la comunicazione col software produttore liberando risorse. Saranno perciò da prevedere degli automatismi/crontab nell’esecuzione di questi task.

In considerazione che il formato dell’XML (XMLSIP) utilizzato per la codifica dei metadati in ingresso a TAKo potrebbe subire delle revisioni nel corso tempo in funzione del variare delle esigenze, si ritiene opportuno introdurre un sistema di “versionamento” delle specifiche stesse (per altro già previsto dallo standard Parer). Ad ogni revisione del formato verrà assegnato uno specifico codice di versione che identifica una data revisione, tale valore dovrà essere riportato nell’XML. Il processo di “parsing”, analizzando il codice di versione, sarà in grado di comprendere a quale versione fa riferimento l’XML ricevuto attivando così il corretto processo di decodifica. In questo modo si intende garantire una certa retro-compatibilità nei confronti dei software produttori.

5.5.1 Processo parsing

Il diagramma illustra una sintesi del processo di parsing/eleaborazione dei dati in input a TAKo:



- Il software produttore genera il pacchetto SIP (XML metadati + files) e lo inoltra al sistema TAKo
- TAKo riceve il pacchetto SIP, estrae i file (li memorizza in area temporanea) e l'XML dei metadati (XMLSIP – descrizione unità documentaria)
- Viene verificata la correttezza formale dell'XMLSIP usando l'XSD corretto (l'XSD deve corrispondere alla medesima versione delle specifiche usate per redigere l'XMLSIP); in caso di difformità, o versione delle specifiche XML non più supportate, viene restituito un codice di errore
- Viene individuata la versione delle specifiche usate per la redazione dell'XMLSIP, viene caricato il parser corretto per la specifica versione (UnitDocParserV.XXX)
- Esecuzione del processo di parsing, l'XMLSIP viene convertito in una struttura dati elaborabile dal software TAKo, un “oggetto informatico” (programmazione ad oggetti) denominato “UnitaDocObjectV.XXX”, tale oggetto gestisce le medesime informazioni dell'XMLSIP (è una sua diversa rappresentazione) ed è perciò anch'esso legato alla specifica versione dell'XMLSIP ricevuto
- Viene verificato se l'oggetto “UnitaDocObjectV.XXX” appartiene all'ultima versione dello standard, in caso contrario viene eseguita una conversione al fine di ottenere l'oggetto “UnitaDocObjectV.LAST” (da analizzare come gestire eventuali errori di conversione)
- Viene verificata la struttura dell'unità documentaria ricevuta:
 - il sistema verifica che siano presenti tutti i documenti, componenti, sottocomponenti previsti per la specifica tipologia di unità documentaria (confronto con catalogo struttura presente in DB TAKo)
 - Viene verificata la correttezza dei metadati specifici (non fissati a livello di standard) che dipendono dalle tipologie (unità documentaria, documento, componente/sotto-componente)
 - In caso di errori, viene restituito un codice di errore
- Vengono verificati i file ricevuti, controllo presenza e impronte (hash dichiarati in XML e hash ricalcolati da TAKo); in caso di errori il sistema restituisce un codice di errore
- I metadati gestiti dall'oggetto “UnitaDocObjectV.LAST” vengono convertiti in una struttura dati di tipo JSON
- Le informazioni vengono memorizzate stabilmente nel sistema TAKo:
 - vengono indicizzati i metadati utili per future ricerche (nel DB)
 - tutti i metadati (in formato JSON) vengono memorizzati per le future operazioni sul SIP (nel DB)
 - i file permangono nell'area temporanea in attesa dell'invio in conservazione (dovranno essere inviati al conservatore e non gestiti per sempre da TAKo)
- Viene restituita al sistema produttore la conferma della presa in carico dello SIP da parte di TAKo

Figure 3

5.5.2 Processo conversione

Il diagramma illustra una sintesi del processo di conversione/invio in conservazione (output TAKo):

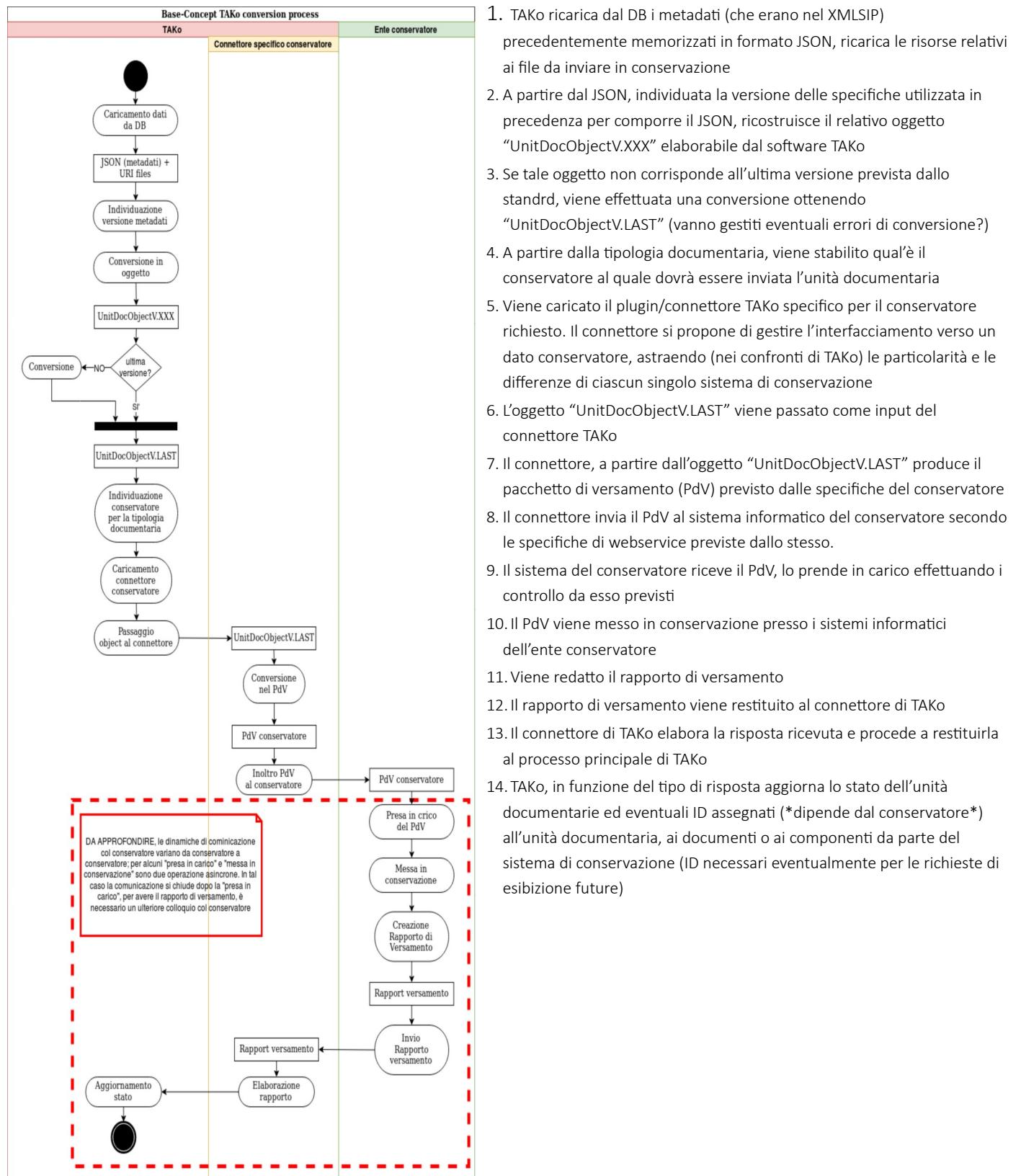


Figure 4

NOTA: lo schema riportato in precedenza, avendo come focus la macro analisi, semplifica la reale e possibile dinamica di comunicazione con un sistema del conservatore (vedi area riquadro rosso).

Nell'esempio precedente la presa in carico del PdV, la sua messa in conservazione e la produzione del rapporto di conservazione sono eseguite in modo sincrono all'interno di un'unica operazione atomica. Conseguentemente, al termine dell'upload del PdV, il connettore di TAKo ottiene immediatamente l'esito della procedura di messa in conservazione.

Esistono però degli enti conservatore i cui sistemi prevedono che tali operazioni siano effettuate in tempi distinti (asincrone), in tal caso è perciò necessario che il connettore TAKo preveda due distinte chiamate verso il conservatore:

1. invio del PdV al conservatore, si tratta di una semplice presa in carico, il conservatore risponde al connettore TAKo con l'esito della presa in carico
2. richiesta stato conservazione, il connettore TAKo periodicamente interroga il sistema del conservatore per ottenere lo stato della messa in conservazione; in caso di conservazione avvenuta (o errore) procede ad aggiornare lo stato dell'unità documentaria

In fase di progettazione sarà necessario dettagliare questi aspetti maggiormente e prevedere degli automatismi/crontab che possano gestire queste dinamiche adeguatamente.

E' inoltre da tenere in considerazione che, per alcuni sistemi di conservazione, sarà necessario invocare il processo di "esibizione" (il processo col quale il conservatore restituisce i documenti) solo utilizzando gli identificativi assegnati dal conservare stesso. Di norma questi ID, quando previsti, sono restituiti dal conservare stesso all'interno del rapporto di versamento, nell'elaborare il rapporto di versamento, e aggiornamento dell'unità documentaria, sarà necessario considerare anche questa dinamica.

5.6 Parser, UnitDocObject, connettore e converter

I seguenti paragrafi raccolgono idee e spunti inerenti le logiche e la struttura da implementare nella realizzazione dei seguenti componenti del software TAKo che verranno usati nei processi di invio in conservazione.

TAKo verrà sviluppato mediante tecniche di programmazione orientata agli oggetti, si tratta di un paradigma di programmazione che permette di definire degli “oggetti software” in grado di interagire gli uni con gli altri attraverso lo scambio di messaggi. La programmazione ad oggetti fornisce un supporto naturale alla modellazione software, gli oggetti software rappresentano sostanzialmente il corrispettivo degli oggetti del mondo reale.

All’interno del software TAKo verranno definiti differenti oggetti:

- Parser (“UnitDocParser”) – l’oggetto informatico incaricato di validare, leggere ed estrarre il contenuto dell’XMLSIP ricevuto, ricevendo come input il pacchetto SIP (in particolare l’XMLSIP) produrrà una istanza dell’oggetto “UnitDocObject”
- XMLSipIndex – l’oggetto che codifica la struttura dell’XMLSIP, si tratta di un “indice” mediante il quale è possibile stabilire i tag presenti nell’XML e il datatype (tipo di dato, string, intero...) ad essi associati. Viene utilizzato dall’oggetto parser per poter correttamente elaborare l’XML
- “UnitDocObject” – è l’oggetto prodotto come output del parser che costituisce la rappresentazione informatica dell’SIP ricevuto in ingresso da TAKo, in particolare contiene le informazioni che erano memorizzate in origine nell’XMLSIP
- “connettore/plugin” - si tratta di un componente modulare di TAKo sviluppato per permettere al sistema di interfacciarsi con un determinato ente conservazione (invio PdV, recupero PdA, controllo stato PdV,...), per ogni sistema di conservazione verrà sviluppato uno specifico connettore/plugin; durante la fase di versamento il “connettore” riceverà in input una istanza dell’oggetto “UnitDocObject” e, a partire da essa, produrrà ed invierà il PdV al conservatore
- Converter – oggetto informatico, sviluppato all’interno di uno specifico connettore/plugin (connettore verso sistema di conservazione), che permette la creazione del PdV secondo le specifiche previste dal conservatore, riceve come input una istanza dell’oggetto “UnitDocObject” e produce come output l’XML da inserire nel PdV per la rappresentazione dei metadati (o altro formato previsto dalle specifiche del conservatore)

Come già riportato nell’introduzione del presente documento, al momento non è previsto uno standard di versamento a livello nazionale/internazionale, conseguentemente ogni ente conservatore ha realizzato il proprio sistema informatico secondo le proprie logiche.

La realizzazione in TAKo di una architettura a “plugin” (nella sua componente di connessione verso i vari conservatori) ha come obiettivo:

- realizzare un sistema “espandibile” che permetta il supporto ai differenti conservatori ad oggi operativi (o ad eventuali nuovi enti conservatori)
- isolare le problematiche e dinamiche di versamento inerenti uno specifico conservatore all’interno dello specifico connettore/plugin

- sviluppare le funzionalità richieste a TAKo, strutturando in modo “centralizzato” tutte quelle componenti che invece sono trasversali, comuni e non dipendono perciò ad uno specifico ente conservatore

Nonostante l'assenza di uno standard, al momento, la gran parte degli enti conservatori addotta un file XML per la codifica dei metadati associati all'unità documentaria/documenti che devono essere inviati in conservazione.

Le maggiori differenze si hanno a livello logico, la struttura/modello dati dei vari pacchetti di versamento sono differenti da conservatore a conservatore:

- **Parer:** per strutturare il suo PdV (SIP) è previsto un completo e complesso modello dati che permette di rappresentare una unità documentaria in modo completo ed esaustivo
- **Arancia:** il PdV non si prefigge di rappresentare una unità documentaria, ma un insieme di file/documenti aggregati per mezzo di un file ZIP (lo zip è il pacchetto di versamento)
- **Aruba:** il PdV non si prefigge di rappresentare una unità documentaria, è prevista una struttura a cartelle (PdA), ciascuna cartella può contenere più PdV, ogni PdV è una aggregazione di file/documenti
- **LegalDoc:** il PdV non si prefigge di rappresentare una unità documentaria, inoltre è possibile inviare un solo file per ciascun PdV, in questo caso il PdV non è considerabile come aggregazione

Vi sono perciò due principali problematiche che dovranno essere risolte da TAKo:

- la conversione dell'XML utilizzato per la codifica dei metadati, dal formato di Input (di TAKo, il pacchetto SIP) al formato di output (PdV) previsto dallo specifico ente conservatore
- la “mappatura” del modello dati previsto da TAKo (unità documentaria) al modello previsto dal conservatore per il PdV

A ciascun singolo plugin, in aggiunta alla gestione delle fasi di comunicazione con il sistema informatico dell'ente conservatore, saranno perciò assegnati i seguenti compiti:

- produrre l'XML secondo quanto previsto dalle specifiche dell'ente conservatore, si dovrà predisporre un modello/sistema che permetta di rimappare i campi dinamicamente, dalla struttura dell'XML in input (standard TAKo) alla specifica struttura dell'XML di output (standard conservatore)
- “mappare” il modello dati di TAKo (unità documentaria) nel corrispondente modello dati previsto per il PdV del conservatore garantendo in tal modo il corretto invio in conservazione:
 - **Parer:** il modello dati di TAKo e Parer coincidono, non vi dovrebbero essere sostanziali problemi
 - **Arancia:** il PdV si basa su una aggregazione di documenti mediante uno ZIP, si potrebbe considerare lo ZIP coincidente ad una unità documentaria
 - **Aruba:** il PdV si basa su una aggregazione logica di documenti, si potrebbe considerare il PdV corrispondente ad una singola unità documentaria; va inoltre tenuto in considerazione che Aruba permette di recuperare un intero pacchetto solo a partire dall'ID assegnato ad un PdA (una cartella di più PdV), se si desidera poter recuperare puntualmente una unità documentale è necessario creare dei PdA contenenti un solo PdV (anche sembra una cosa sconsigliata da Aruba, da analizzare questo aspetto)

- **LegalDoc**: ogni PdV permette di inviare un singolo file, l'unica soluzione sembra inviare tanti PdV quanti sono i file presenti nell'unità documentale (a TAKo rimarrebbe così l'onere di mantenere l'aggregazione)
- restituire a TAKo le informazioni per “associare” l'unità documentaria indicizzata (in TAKo) con il corrispondente PdA (pacchetto di archiviazione) che è stato creato dal conservatore nel suo sistema di conservazione al termine del processo di messa in conservazione; questa associazione è indispensabile per poter garantire la possibilità di recuperare (esibire) dal sistema di conservazione l'intera unità documentare assieme ai suoi allegati:
 - **Parer**: permette di recuperare sia l'intera unità documentaria, che il singolo documento; gli identificativi per il recupero coincidono con quelli assegnati dal software produttore (ossia la chiave del SIP per l'unità documentaria, o chiave SIP+IDDocumento per il singolo documento)
 - **Arancia**: permette di recuperare solo l'intero PdV (lo ZIP) a partire dall'ID assegnato dal sistema di conservazione di arancia
 - **Aruba**: permette di recuperare il singolo file o l'intera cartella di PdV (ossia il PdA) utilizzando gli identificativi assegnati dal sistema informatico di Aruba; per questa motivazione è da valutare se associare ad un PdA un solo PdV, e al PdV una sola unità documentaria
 - **LegalDoc**: permette di recuperare un solo file alla volta, dato che un PdV corrisponde al singolo file, il recupero avviene solamente utilizzando l'ID assegnato da LegalDOC

In capo al “core” di TAKo rimarranno le operazioni di:

- parsing del SIP e indicizzazione della corrispondente unità documentale
- monitoraggio dello stato (d'invio) di ciascun SIP (conservato/errore/ancora da inviare....)
- associazione tra unità documentaria ed elementi identificativi dal PdA lato conservatore

Nella pagina che segue sono riportati due XML raffiguranti un SIP TAKo (lato sinistro) e un PdV (lato destro) previsto dal sistema di conservazione DocFly di Aruba.

I concetti che saranno riportati nei paragrafi che seguiranno (parser/converter...) si baseranno su questo esempio.

L'esempio riporta un SIP prodotto da un sistema documentale, ed inviato a TAKo, per l'invio in conservazione di una fattura attiva.

L'immagine evidenzia le operazioni di ri-mappatura che dovranno essere previste dal sistema, sia TAKo, che il singolo connettore/plugin dovranno essere configurabili per permettere di gestire questa complessità.

I metadati aggregati secondo lo standard TAKo/Parer (unità documentaria, documento, componente) specifici o “fissi”, dovranno essere ri-mappabili e raggruppabili per rispettare il PdV del conservatore.

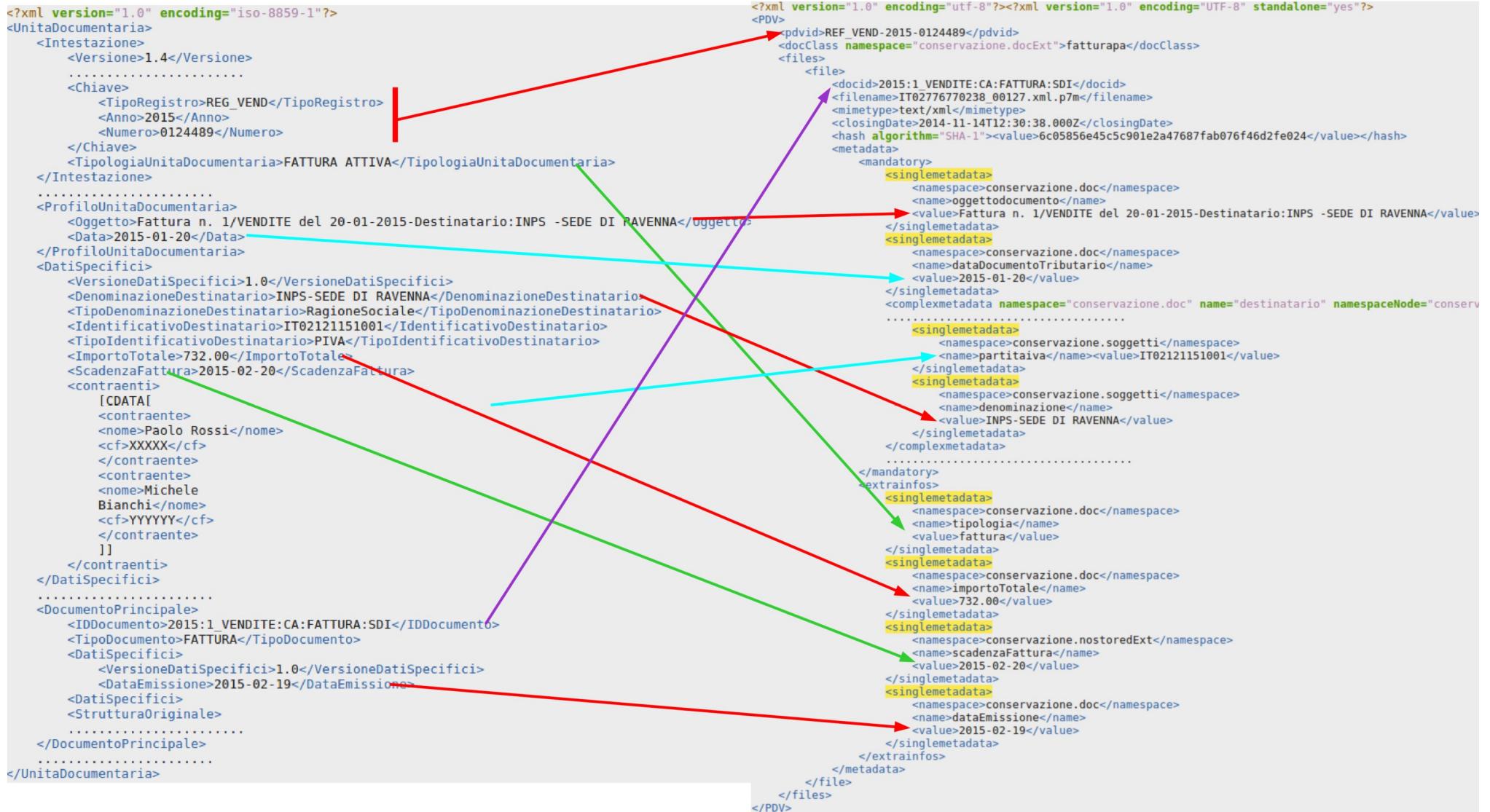


Figure 5

5.6.1 UnitDocObject

L'UnitDocObject è l'oggetto informatico che si propone di rappresentare e modellare all'interno del software TAKo il pacchetto SIP ricevuto in ingresso ed in particolare le informazioni memorizzate all'interno dell'XML dei metadati (XMLSIP).

Il modello dati espresso nel paragrafo "Macro struttura SIP Parer", che deriva dall'analisi delle specifiche Parer e relativo XMLSIP, rappresenta il punto di partenza per la definizione della struttura da assegnare all'oggetto "UnitDocObject".

Analizzando tali specifiche, e l'XMLSIP, è possibile evidenziare che l'elemento minimo rappresentabile è un singolo tag XML, es:

```
<TipoRegistro id="XXX" >VEND</TipoRegistro>
```

Tale tag possiede:

- un nome
- uno o più attributi valorizzati (es. "id"; al momento però lo standard Parer non prevede l'uso di attributi)
- un valore associato al tag stesso (es. REG_VEND)

Considerando l'XML e il tag da un punto di vista "informatico", la variabile "TipoRegistro" memorizza un dato di tipo "string" (datatype) valorizzato con il valore "REG_VEND".

Tali tipologie di variabili costituiscono perciò l'elemento minimo rappresentabile all'interno dell'oggetto "UnitDocObject".

Ogni elemento/tag dell'XML può essere così rappresentato per mezzo di un "ElementObject", ossia un "oggetto informatico" che al suo interno memorizzi le informazioni memorizzate in origine nel tag XML:

- nome del tag
- attributi e relativi valori
- valore memorizzato nel tag

I valori memorizzati nel tag (o negli attributi) possono appartenere ad uno dei seguenti "datatype" (nella successiva fase di analisi potranno essere individuati altri datatype):

- stringa (test)
- intero
- decimale
- data
- timestamp (data + orario)
- orario

Il valore memorizzato nel tag sarà memorizzato nell'oggetto “ ElementObject” mediante l'oggetto nativo, previsto per la codifica di un determinato datatype, dal linguaggio di programmazione col quale TAKo sarà sviluppato.

L'XML permette di raggruppare i vari tag al fine di ottenere aggregazioni logicamente coerenti di informazioni appartenenti ad medesimo sotto-dominio dell'unità documentale:

```
<Chiave>
  <TipoRegistro>REG_VEND</TipoRegistro>
  <Anno>2015</Anno>
  <Numero>0124489</Numero>
</Chiave>
```

Il risultato è una corretta organizzazione delle informazioni all'interno dell'XML, si crea conseguentemente un rapporto padre-figlio tra i vari elementi.

L'oggetto “ElementObject” dovrà perciò esser rivisto per poter codificare queste relazioni, verranno aggiunte due variabili:

- parent, la variabile memorizzerà un puntatore all' ElementObject padre (es. TipoRegistro.parent=Chiave)
- children, dizionario contenente gli elementi figli (es. in Chiave, children={‘TipoRegistro’:TipoRegistro, ‘Anno’:Anno, ‘Numero’:Numero})

L'oggetto “ElementObject” dovrà essere dotato di funzioni che permettano di (funzioni da rivedere in fase di progettazione):

- impostare il valore di un attributo (fase di parsing)
- recuperare l'elenco degli attributi presenti
- recuperare il valore di uno specifico attributo
- impostare il valore dell'elemento (fase di parsing)
- recuperare il valore dell'elemento
- recuperare l'elenco degli elementi figli presenti
- recuperare un ElementObject figlio
- convertire l'ElementObject) in una sua rappresentazione JSON (fase stoccaggio UnitDocObject in DB TAKo)
- convertire una rappresentazione JSON (di ElementObject) nell' ElementObject (fase caricamento dati da DB TAKo e ricostruzione UnitDocObject)
- recupero di un valore o attributo di un elemento per mezzo di una sintassi del tipo XPATH (funzionalità per la mappatura dai metadati da SIP a PdV effettuata dallo specifico connettore)

L'oggetto UnitDocObject potrebbe quindi estendere l'oggetto (la classe) ElementObject aggiungendo (funzioni da rivedere in fase di progettazione) che permettano di accedere ad esempio ad determinate informazioni:

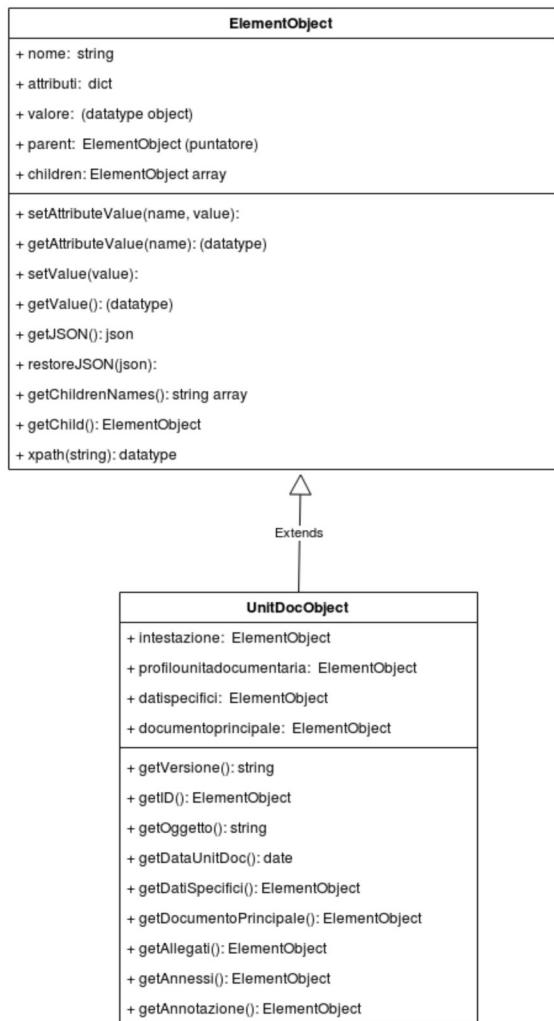
- versione sip
- Id dell'unità documentaria (chiave)
- Oggetto dell'unità documentaria
- Data dell'unità documentaria
- Dati specifici dell'unità documentaria
-
- Documento principale

- Allegati
- Annessi
- Annotazioni

Nella pagina che segue sono riportati due schemi:

- ClassDiagram, una possibile struttura (variabili e funzioni) della classe di appartenenza dell'oggetto ElementObject e UnitDocObject
- Una mappa concettuale che illustra la struttura di un UnitDocObject e l'annidamento dei vari ElementObject che lo vanno a comporre

ClassDiagram (concept)



MindMap

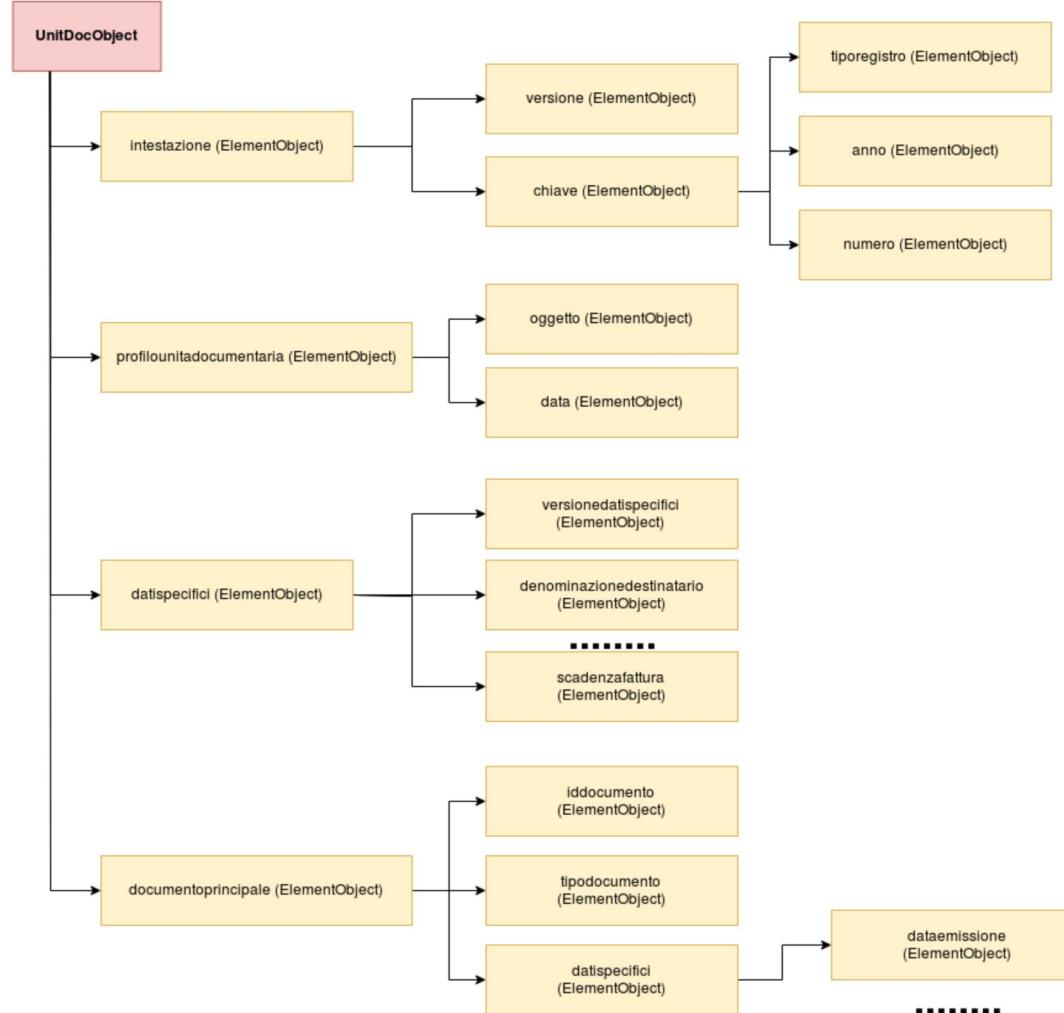


Figure 6

5.6.2 Parser ed XMLSipIndex

Il parser (“UnitDocParser”) è l’oggetto informatico incaricato di validare, leggere ed estrarre quanto contenuto all’interno dell’XMLSIP, (l’XML ricevuto all’interno del pacchetto SIP ed utilizzato per descrivere e trasmettere i metadati dell’unità documentaria).

L’UnitDocParser riceve in input l’XMLSIP, ne effettua la verifica formale utilizzando l’XSD di riferimento e procede con l’operazione di estrazione (parsing) delle informazioni codificate nell’XML. Il parser produce un oggetto “UnitDocObject” come output della procedura di parsing, tale oggetto memorizza perciò le informazioni che in origine erano presenti nell’XML e associate al SIP.

L’XML, essendo sostanzialmente un formato testuale, memorizza al suo interno tutte le informazioni come semplice stringhe di testo. L’UnitDocObject dovrà invece, per permettere una corretta elaborazione dei metadati, memorizzare tali informazioni codificandole nel corretto datatype. Il parser dovrà perciò essere in grado di stabilire, per ogni singolo tag dell’XML, la natura del dato (datatype) in esso memorizzato al fine di effettuare una corretta “conversione” (casting) dalla stringa testuale dell’XML al datatype previsto in UnitDocObject.

L’XMLSIP prevede sostanzialmente due tipologie di metadati:

- “fissi”, non dipendono dalla tipologia dell’unità documentaria, dalla natura del documento o componente; sono codificati mediante il file XSD associato all’XMLSIP
- dinamici (dati specifici), dipendono dalla tipologia documentaria, dalla tipologia del documento e dalla tipologia del componente

Mentre i metadati fissi non variano sostanzialmente nel tempo (se non al variare delle specifiche del pacchetto SIP), quelli dinamici, oltre ad essere differenti in funzione delle tipologie, possono variare molto più frequentemente anche nel tempo:

- nascita di nuove tipologie di unità documentarie
- nascita di nuove tipologie di documento
- nascita di nuove di tipologie di componente
- aggiunta o rimozione di uno specifico metadato associato ad una data tipologia

L’esempio che segue illustra una possibile sezione “dati specifici”:

```

<DatiSpecifici>
  <VersioneDatiSpecifici>1.0</VersioneDatiSpecifici>
  <DenominazioneDestinatario>INPS - SEDE DI RAVENNA</DenominazioneDestinatario>
  <TipoDenominazioneDestinatario>RagioneSociale</TipoDenominazioneDestinatario>
  <IdentificativoDestinatario>PIVA</IdentificativoDestinatario>
  <ImportoTotale>732.00</ImportoTotale>
  <ScadenzaFattura>2015-02-20</ScadenzaFattura>
  <contraenti>
    [CDATA[
      <contraente>
        <nome>Paolo Rossi</nome>
        <cf>XXXXXX</cf>
      </contraente>
      <contraente>
        <nome>Michele
        Bianchi</nome>
        <cf>YYYYYY</cf>
      </contraente>
    ]]
  </contraenti>
</DatiSpecifici>

```

Figure 7

è così possibile individuare due tipologie di metadati specifici:

- semplici (es. “ScadenzaFattura”), il tag memorizza un semplice intero, stringa testuale o data, l’informazione è sostanzialmente riconducibile ad un datatype “standard”, in questo caso il parser può agevolmente convertire (casting) la stringa XML nel datatype corretto
- complessi (es. “contraenti”), il tag memorizza una struttura complessa, non codificata nell’XSD, con ulteriori sotto-campi strutturati, possono essere considerati come dei datatype “custom/speciali”, in questo caso il parser, senza informazioni aggiuntive, non è in grado di effettuare un classico casting; da tenere inoltre in considerazione che i metadati “complessi” vengono codificati in una sezione “CDATA” o mediante caratteri di escape, risulta pertanto necessaria anche una ulteriore preventiva operazione di conversione (da CDATA o escape)

Si prevede pertanto, all’interno di TAKo, di realizzare un catalogo di tutti i datatype (semplici e complessi) che possono essere utilizzati per la codifica dei metadati specifici all’interno dell’XML.

Ad ogni singolo datatype verrà associata una specifica funzione di “parsing+casting” che permetta di estrarre correttamente le informazioni dall’XML per un dato tag e produrre in output un “ElementObject” correttamente codificato:

- string => STR_PARSER()
- integer => INT_PARSER()
- date => DATE_PARSER()
- float => FLOAT_PARSER()
- contraenti => CONTRAENTI_PARSER()

Questo tipo di associazione può essere effettuata a livello di codice, nella definizione della classe per l’UnitDocParser.

A questo punto risulta evidente la necessità di disporre di un “indice/catalogo” (XMLSipIndex) di tutti i possibili tag che si possono presentare all’interno dell’XMLSIP, tale indice permetterà di associare ad ogni singolo tag dell’xml il suo corretto datatype e, conseguentemente, permetterà al parser di stabilire la corretta funzione di “parsing+casting” da invocare, per esempio:

- “DenominazioneDestinatario” => string => STR_PARSER()
- “ImportoTotale” => float => FLOAT_PARSER()
- “contraenti” => contraenti => CONTRAENTI_PARSER()

Il parser, nell’elaborare il tag “ImportoTotale”, interrogherà l’indice “XMLSipIndex” per stabilire il datatype associato al tag, trattandosi di un “float” invocherà la funzione di parsing “FLOAT_PARSER()”. La funzione “FLOAT_PARSER” restituirà un oggetto “ElementObject” nel quale il valore “732,00” sarà codificato come datatype “numero decimale”, l’oggetto “ElementObject” sarà perciò inserito dal parser nella corretta posizione all’interno dell’oggetto “UnitDocObject” (vedi precedente “MindMap”).

Nella definizione dell’ XMLSipIndex è necessario tenere in considerazione che i metadati codificati all’interno dell’XMLSIP possono essere sia fissi che dinamici (dati specifici).

I metadati fissi possono perciò essere catalogati/associati al rispettivo datatype all’interno di XMLSipIndex direttamente a livello di codice. Bisogna tener comunque in considerazione che anche i “metadati fissi” in realtà possono variare da versione a versione delle specifiche SIP. Per questo sarà necessario definire, per ogni versione delle specifiche SIP, un specifico XMLSipIndex.

Per quanto riguarda i metadati dinamici (dati specifici), essendo prevista una possibile frequente variazione nella loro struttura, non è possibile “codificarli” all’interno del codice dell’XMLSipIndex, è necessario predisporre una modalità di configurazione che non preveda l’intervento di un programmatore e l’alterazione del codice sorgente di TAKo.

Da tenere inoltre in considerazione che anche le sezioni “DocumentoPrincipale”, “Allegati”, “Annessi” e “Annotazioni” risultano essere “dinamiche”. Le tipologie di documenti che possono essere associati ad una unità documentale dipendono infatti dalla tipologia della stessa unità documentaria; a sua volta in funzione della tipologia di documento sono previsti determinati componenti o meno.

Si prevede pertanto la realizzazione di apposite strutture dati nel database di TAKo che permettano la codifica di tutta questa complessità logica associando inoltre, ad ogni metadato dinamico, il corrispondente datatype.

In questo modo, grazie alla realizzazione di adeguate interfacce d’amministrazione, la definizione delle tipologie, della struttura dei documenti e dei relativi metadati specifici, può essere fatta agevolmente dall’amministratore di sistema di TAKo.

In considerazione di quanto sopra riportato le informazioni gestite da XMLSipIndex risultano essere dinamiche, l’indice complessivo dei metadati restituito da XMLSipIndex varia al variare della tipologia documentaria la quale sostanzialmente stabilisce la possibile struttura dei documenti e i metadati dinamici.

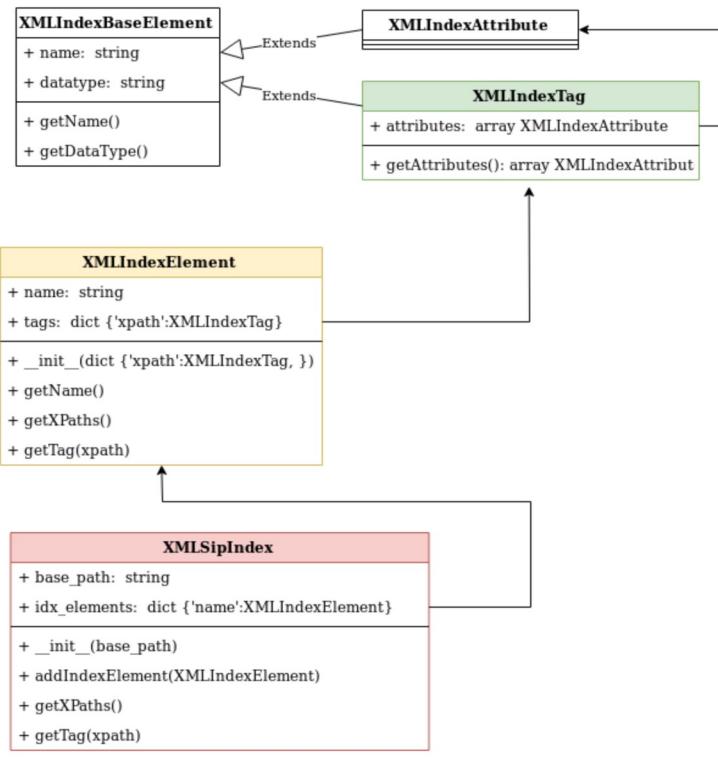
XMLSipIndex dovrà pertanto “fondere” l’indice dei metadati fissi (realizzato a livello di codice) con l’indice dei metadati dinamici ottenuto a partire da una query effettuata sul DB in funzione della tipologia documentaria.

L’indice così ottenuto, oltre ad essere necessario durante la fase di parsing, sarà utilizzato anche nelle successive fasi di invio al conservatore dallo specifico connettore/pluging per la configurazione del proprio converter, il “catalogo” di tutti i metadati in input permetterà di stabilire la ri-mappatura verso i metadati del

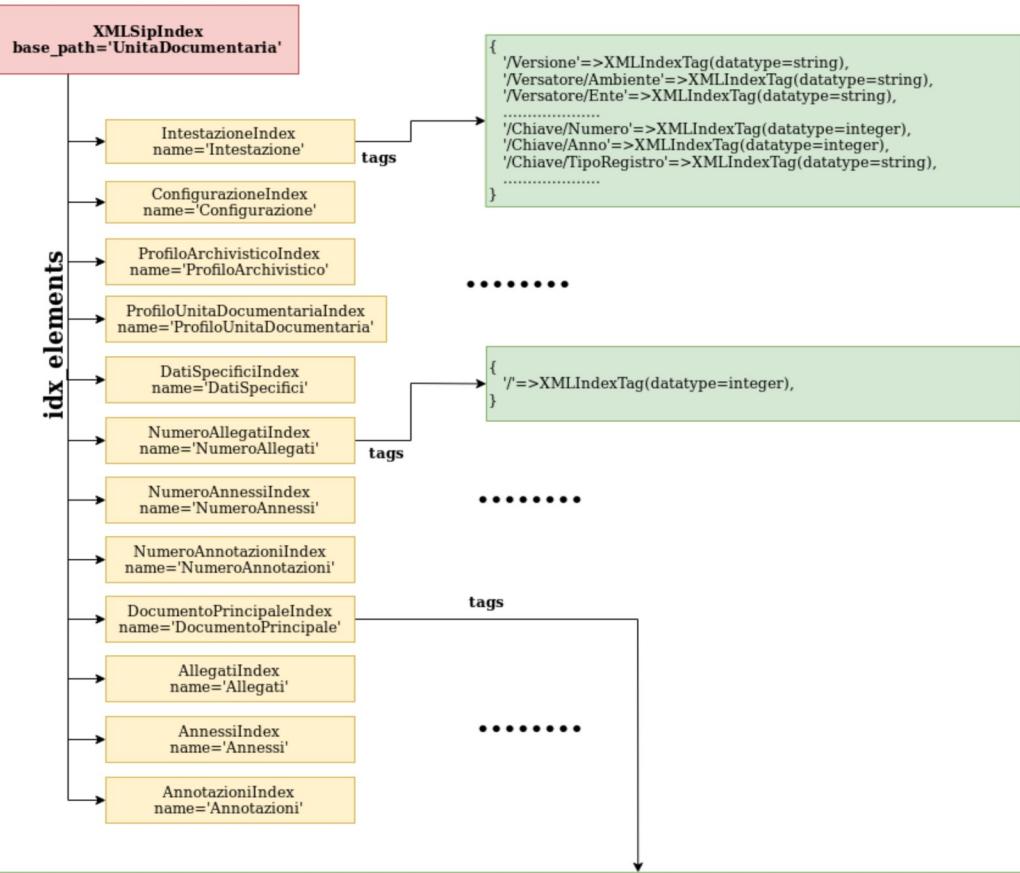
PdV di output anche dinamicamente (l'amministratore di TAKo potrà modificare la configurazione del connettore/plugin, fare riferimento al successivo paragrafo).

La pagina che segue illustra la possibile struttura della classi che si andrà a realizzare per la definizione dell'XMLSipIndex e una mappa concettuale di come le informazioni verranno memorizzate.

ClassDiagram (concept)



MindMap



```

{
  '/[TipoDocumento=FATTURA]/IDDocumento'=>XMLIndexTag(datatype=string),
  '/[TipoDocumento=FATTURA]/TipoDocumento'=>XMLIndexTag(datatype=string),
  '/[TipoDocumento=FATTURA]/DatiSpecifici/VersioneDatiSpecifici'=>XMLIndexTag(datatype=string),
  '/[TipoDocumento=FATTURA]/DatiSpecifici/DataEmissione'=>XMLIndexTag(datatype=date),
  '/[TipoDocumento=FATTURA]/StrutturaOriginale/Componenti/Componente[TipoComponente=Contenuto]/NomeComponente'=>XMLIndexTag(datatype=string),
  '/[TipoDocumento=FATTURA]/StrutturaOriginale/Componenti/Componente[TipoComponente=Contenuto]/SottoComponenti/SottoComponente[TipoComponente=FoglioDiTrasformazione]/ID'=>XMLIndexTag(datatype=string),
  .....

  '/[TipoDocumento=ACCONTO]/IDDocumento'=>XMLIndexTag(datatype=string),
  '/[TipoDocumento=ACCONTO]/TipoDocumento'=>XMLIndexTag(datatype=string),
  '/[TipoDocumento=ACCONTO]/DatiSpecifici/VersioneDatiSpecifici'=>XMLIndexTag(datatype=string),
  '/[TipoDocumento=ACCONTO]/DatiSpecifici/DataEmissione'=>XMLIndexTag(datatype=date),
  '/[TipoDocumento=ACCONTO]/StrutturaOriginale/Componenti/Componente[TipoComponente=Contenuto]/NomeComponente'=>XMLIndexTag(datatype=string),
}
  
```

Figure 8

Le informazioni relative ad un singolo tag (datatype del dato, attributi previsti e relativi datatype) verranno memorizzate mediante oggetti di tipo:

- XMLIndexAttribute – per la gestione dell’informazioni di un singolo attributo al tag (nome attributo e datatype)
- XMLIndexTag – per la gestione dell’informazioni di un singolo tag, conterrà (nome tag, datatype, tanti XMLIndexAttribute quanti gli attributi previsti [array])

L’XMLSipIndex permette di recuperare le informazioni relative ad uno specifico tag dell’XMLSIP mediante delle interrogazioni che utilizzano una sintassi di tipo XPATH:

- /UnitaDocumentaria/Intestazione/Versione => restituirà le informazioni (l’oggetto XMLIndexTag) del tag fisso “Versione” presente all’interno dell’elemento “Intestazione”
- /
UnitaDocumentaria/DocumentoPrincipale[TipoDocumento=FATTURA]/StrutturaOriginale/Componenti/Componente[TipoComponente=Contenuto]/NomeComponente => restituirà invece le informazioni del tag “NomeComponente” che appartiene al tag “Componente” se TipoComponente=Contenuto, del documento principale se il documento principale è un documento di tipo “FATTURA”
- /
UnitaDocumentaria/DocumentoPrincipale[TipoDocumento=FATTURA]/DatiSpecifici[VersioneDatiSpecifici=1.0]/ImportoTotale => restituirà le informazioni del tag “ImportoTotale” che è un “metadato dinamico” del documento principale di tipo FATTURA, previsto per la versione “1.0” dei relativi dati specifici.

Come visibile dalla “MindMap” l’XMLSIP è stato scomposto nelle sue principali sezioni (elementi gialli):

- Intestazione
- Configurazione
- ProfiloArchivistico
- ProfiloUnitàDocumentaria
- DatiSpecifici
- NumeroAllegati
- NumeroAnnessi
- NumeroAnnotazioni
- DocumentoPrincipale
- Allegati
- Annessi
- Annotazioni

Ogni “macro sezione” viene rappresentata mediante un oggetto di tipo “XMLIndexElement” il quale cataloga al suo interno tutti i tag previsti (per quella macro sezione) e ne memorizza il nome (della macro sezione, name).

Ogni XMLIndexElement memorizza tanti oggetti “XMLIndexTag” quanti sono i suoi sotto tag . L’insieme dei XMLIndexTag è organizzato mediante un dizionario (tags) nel quale ad ogni XMLIndexTag è associata una chiave univoca “XPATH”.

L’insieme delle sezioni (XMLIndexElement) viene memorizzato all’interno dell’oggetto XMLSipIndex che costituisce l’interfaccia principale per la consultazione dell’indice. Ogni XMLIndexElement verrà aggiunto al suo interno mediante la funzione “addIndexElement” che provvederà a catalogare correttamente la sezione nel dizionario “idx_elements”.

A causa della natura fissa della struttura di alcune sezioni di metadati e dinamica di altre:

- le sezioni “fisse” saranno codificate direttamente all’interno del codice sorgente di TAKo (e varieranno solo al variare della versione delle specifiche del SIP), es. “IntestazioneIndex”
- le sezioni “dinamiche”, es. DocumentoPrincipale o DatiSpecifici (dell’unità documentaria), dovranno essere prodotte a partire da una query del database di TAKo in funzione della tipologia di unità documentaria

L’indice potrà essere perciò costruito solo se si è già stabilita quale unità documentaria si sta trattando.

In considerazione di quanto emerso finora, sarà necessario prevedere una procedura “costruttrice” dell’oggetto XMLSipIndex che, a partire dalla tipologia di unità documentaria:

- effettuerà le query per la definizione delle informazioni dinamiche
- creerà gli XMLIndexElement associati a sezione dinamica utilizzando le informazioni ottenute dalle precedenti query (da tenere in considerazione che anche le sezioni dinamiche possiedono dei tag fissi, questi possono essere gestiti al livello di codice)
- creerà gli XMLIndexElement associati a sezione fisso e codificate a livello di codice sorgente
- procederà alla creazione dell’XMLSipIndex mediante gli XMLIndexElement prodotti in precedenza

La funzione “costruttore” dell’XMLSipIndex, e il codice sorgente contenente le informazioni statiche, dovrà poter gestire il variare anche delle informazioni statiche, che infatti possono variare per la definizione di nuove specifiche del pacchetto SIP.

A questo punto mediante l’ XMLSipIndex sarà possibile:

- ottenere la lista di tutti i tag disponibili (getXPaths), funzione prettamente necessaria al connettore verso un dato ente conservatore per poter configurare correttamente la ri-mappatura dei metadati di input nel PdV di output
- definire quali attributi siano presenti in un dato tag (e relativo datatype)
- stabilire il datatype di un tag al fine di effettuare il corretto parsing (informazione usata dal parser)

XMLSIP presenta una complessità elevata dal fatto che la struttura di determinate sezioni dell’XML (in particolare le sezioni con metadati dinamici) dipende dal valore assunto da un determinato tag presente nell’XML stesso, in particolare:

- la sezione “DatiSpecifici” (dall’unità documentaria) dipende dalla tipologia di unità documentaria

- quali tipologie di documenti principali, annotazioni, allegati ed annessi, possano essere presenti dipende dalla tipologia di unità documentaria
- la tipologia di ciascun documento definisce i relativi dati specifici e quali componenti/sottocomponenti possano essere presenti
- la tipologia del componente (o sottocomponente) stabilisce i “dati specifici” di ciascun componente

Anche il parser incaricato di elaborare tale strutture risulterà complesso come diretta conseguenza della complessità intrinseca prevista per l'XMLSIP.

Al momento della stesura di questa macro-analisi, un parser “a logica ad eventi” (l’apertura e chiusura di un tag/sezione dell’xml scatena l’evento di elaborazione di quel tag) non risulta essere ottimale nell'affrontare la complessità sopra esposta, la corretta elaborazione di una sezione dell’XML potrebbe infatti dipendere dal valore assunto da un tag “precedente” la quale elaborazione (evento) è già conclusa. La corretta elaborazione della sezione “DatiSpecifici” dell’unità documentaria (ma anche di “DocumentoPrincipale”) dipende infatti dal tag “TipologiaUnitaDocumentaria”, tag il cui evento di elaborazione è tra i primi ad avvenire all’interno di un parsing ad eventi.

La disponibilità di un indice completo dei possibili tag, e dei relativi XPATH, offerto da XMLSipIndex rende possibile la realizzazione di parser che estragga i contenuti in funzione degli XPATH possibili:

1. Il parser riceve in input l'XMLSIP
2. estrae puntualmente la versione delle specifiche TAKo/Parer previste per l'XMLSIP
3. estrae puntualmente il valore del campo “ TipologiaUnitaDocumentaria” stabilendo la tipologia di unità documentaria che si sta trattando
4. grazie alle informazioni ottenute al punto precedente è possibile ottenere l’indice completo dei tag e XPATH che si possono incontrare all’interno dell’XMLSIP, viene prodotto l’XMLSipIndex relativo alla tipologia documentaria che si sta trattando e alla versione delle specifiche al quale il SIP appartiene
5. Utilizzando le informazioni dell’indice l’XMLSipIndex (quali campi possono essere presenti ed il rispettivo datatype) il parser può procedere all'estrazione delle informazioni e alla creazione dell’oggetto “UnitDocObject”, ossia l’output del processo di parsing

La realizzazione di un parser con le caratteristiche sopra indicate renderebbe sostanzialmente invariata la struttura del suo codice sorgente al variare delle specifiche del pacchetto SIP. A variare, (in funzione della versione delle specifiche), è infatti l’XMLSipIndex che si comporta così come un sorta di “file” di configurazione del parser stesso.

Si prevede comunque di suddividere le operazioni di parsing in funzioni delle “principali sezioni” previste dall’xml:

- Intestazione
- Configurazione
- ProfiloArchivistico
- ProfiloUnitàDocumentaria

- DatiSpecifici
- NumeroAllegati
- NumeroAnnessi
- NumeroAnnotazioni
- DocumentoPrincipale
- Allegati
- Annessi
- Annotazioni

La realizzazione di “sub-parser”, uno per ciascuna sezione, permette di rendere personalizzabili le operazioni di “parsing-eleborazione” in funzione della specifica sezione dell’XM. Le sezioni dell’XML “fortemente dinamiche” necessitano infatti di procedure di elaborazioni più complesse rispetto alle sezioni “statiche” (es. Intestazione).

Il numero delle “principali sezioni” del XMLSIP potrebbe variare da versione a versione delle specifiche SIP, allo stesso modo anche le logiche di “parsing-eleborazione” di una specifica sezione potrebbero differenziarsi in funzione delle specifiche SIP.

Per risolvere questa complessità si può predisporre che:

- Ogni “sezione principale” dell’XMLSIP possieda più oggetti di tipo “sub-parser”, ossia un oggetto “sub-parser” per ogni differente versione delle specifiche SIP, in questo modo si garantisce logiche di elaborazioni distinte al variare della versione delle specifiche. Ogni sub-parser può essere associato direttamente al corrispondente elemento “XMLIndexElement” dell’XMLSipIndex
- La disponibilità di uno specifico “file” di configurazione (per il parser) per ciascuna delle versioni delle specifiche ; tale file di configurazione elenca sostanzialmente le “sezioni principali” da elaborare e per ciascuna l’esatto “sub-parser” incaricato di elaborarla

Alla luce di quanto emerso ulteriormente le operazioni di parsing potrebbero ricondursi a:

1. Il parser riceve in input l’XMLSIP
2. estrae puntualmente la versione delle specifiche TAKo/Parer previste per l’XMLSIP
3. estrae puntualmente il valore del campo “ TipologiaUnitaDocumentaria” stabilendo la tipologia di unità documentaria che si sta trattando
4. grazie alle informazioni ottenute al punto precedente è possibile ottenere l’indice completo dei tag e XPATH che si possono incontrare all’interno dell’XMLSIP, viene prodotto l’XMLSipIndex relativo alla tipologia documentaria che si sta trattando e alla versione delle specifiche al quale il SIP appartiene
5. nota la versione delle specifiche SIP viene caricato il corretto “file di configurazione” per il parser, vengono così creati i principali processi “sub-parser” che risultano ora associati alle rispettive sezioni dell’XMLSipIndex

6. Ogni “sub-parser” procedere ad eseguire il proprio processo di elaborazione utilizzando le informazioni dell’indice l’XMLSipIndex (quali campi possono essere presenti ed il rispettivo datatype), ogni sub-parser può procedere all’estrazione delle informazioni e alla creazione dell’ElementObject per la propria sezione
7. Il parser aggrega i vari ElementObject, uno per sezione principale, prodotto dai vari “sub-parser”, nell’oggetto “UnitDocObject”, ossia l’output del processo di parsing

Nel seguito è riportato una mappa concettuale per il parser in considerazione di quanto finora riportato.

In questa fase di macro-analisi non si ritiene necessario stabilire una struttura per le classi costituenti il parser, il sub-parser e i relativi file di configurazione, si rimanda questo attività alla fase progettuale.

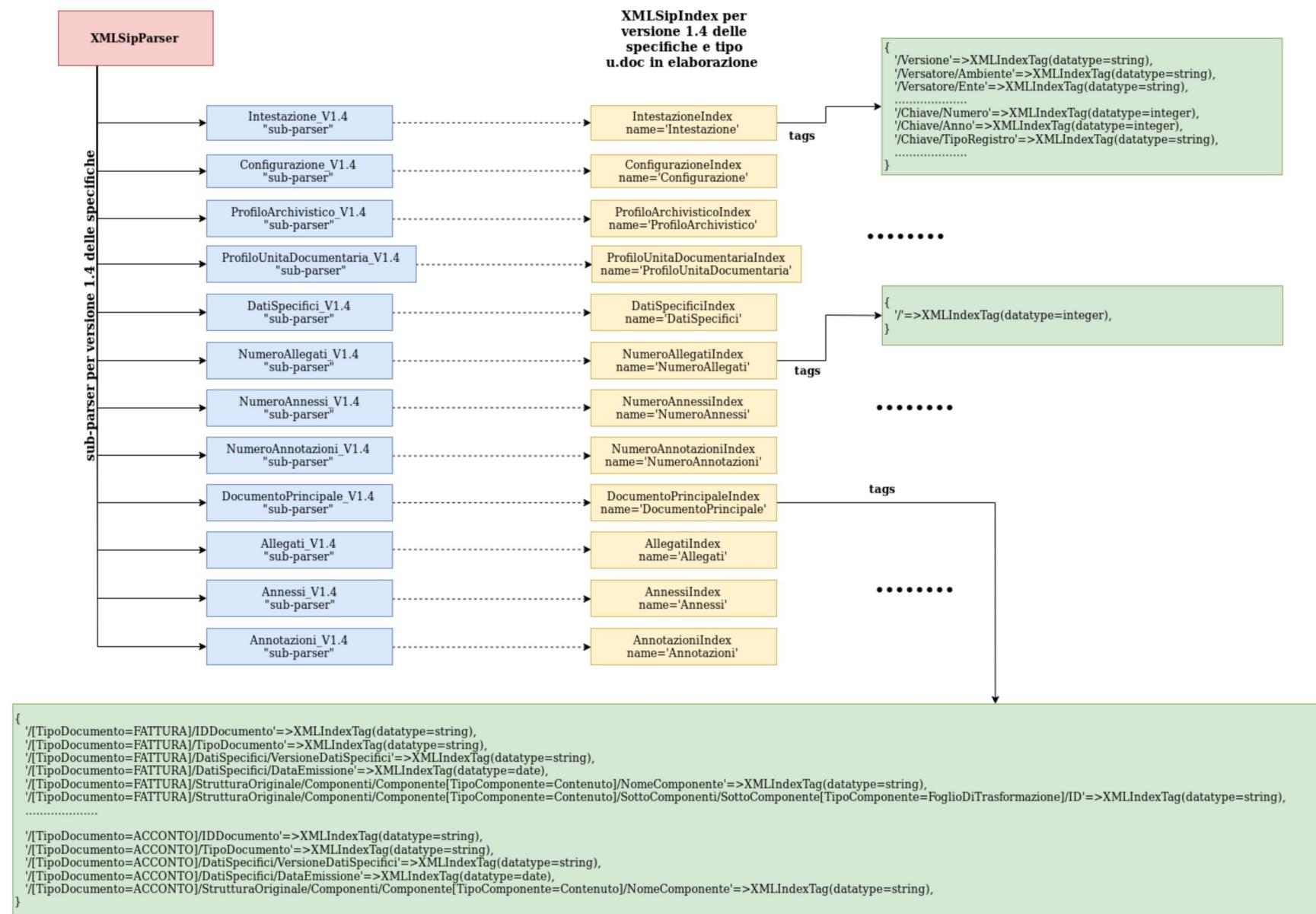


Figure 9

5.6.3 Connettore, converter

Nel presente paragrafo si effettuerà una analisi di massima delle criticità e delle soluzioni che possono interessare la fase di progettazione e sviluppo di un “connettore/plugin” verso DocFly, il connettore che permetterebbe a TAKo di interagire col sistema di conservazione di Aruba spa.

In particolare ci si concentrerà sul processo di invio in conservazione/produzione del PdV da inviare ad Aruba. L'obbiettivo principale è la valutazione dell'intero ciclo di invio in conservazione (produttore=>TAKo=>conservatore), si vuole stabilire la fattibilità, o meno, di realizzare un connettore/plugin a partire dall'infrastruttura TAKo “disegnata/iptizzata” nelle precedenti sezioni del presente documento.

Come esempio pratico si ipotizzerà di dover inviare in conservazione una unità documentaria di tipo “Fattura attiva”, è possibile fare riferimento alla “figura 6” la quale illustra le operazioni di ri-mappatura che dovranno essere effettuate (SIP TAKo => PdV Aruba) da parte del “converter”.

Come già indicato in precedenza:

- il connettore/plugin è un componente modulare di TAKo, è previsto uno specifico connettore/plugin per ciascun ente conservatore supportato da TAKo, è il componente responsabile della gestione di tutte le iterazioni fra TAKo ed il sistema informatico del conservatore nel rispetto delle specifiche tecniche previste da quest'ultimo (le quali variano da conservatore a conservatore)
- converter è il componente interno al connettore/plugin responsabile della conversione dal formato SIP TAKo/Parer al PdV previsto, sostanzialmente produce l'XML dei metadati previsto per lo specifico ente di conservazione

Da una rapida analisi delle specifiche del PdV di aruba si può stabilire che:

- Le specifiche dei metadati di DocFly non si prefiggono di rappresentare una unità documentaria, il punto di vista di Aruba è il singolo file; ogni documento è sostanzialmente costituito da un unico file; ogni pacchetto di versamento (PdV) può contenere più file, cioè più documenti distinti, aventi però in comune la medesima tipologia documentaria (quindi tipologia di unità documentaria)
- ogni pacchetto di versamento (PdV) deve appartenere ad un pacchetto di archiviazione (PdA), il PdA è sostanzialmente una cartella presente sul sistema di aruba che contiene più PdV
- La struttura fornita da Aruba all'XML (che codifica i metadati da inserire nel PdV) non si prefigge come scopo la corretta rappresentazione di un singolo documento o gruppo di documenti. La struttura assegnatagli è stata concepita solo come mero “mezzo/strumento” di trasmissione delle informazioni. L'XSD non prevede un insieme strutturato e ben definito di tag atti alla rappresentazione di una unità documentaria o di un documento, è invece prevista una generica struttura minimale ove, per ogni file/documento, è possibile elencare i metadati ad esso relativi.
- I tag “fissi” stabiliti dall'XSD sono solamente propedeutici alla trasmissione dei file (docid, filename, mimetype, closingdate, hash) e rappresentano l'unico set comune a tutte le tipologie documentarie individuato da Aruba

- Aruba prevede un insieme di classi documentali al fine di classificare la natura dei file che si sta inviando in conservazione, dalla classe documentale derivano i tag previsti nell'xml. Conseguentemente i tag previsti per una certa tipologia documentaria, sia quelli obbligatori (mandatory) che quelli facoltativi (extrainfo), non sono stabiliti a livello di XSD (dato che concepisce solo una generica struttura per l'XML), ma derivano da configurazioni effettuate all'interno del sistema DocFly lato Aruba (vengono create le varie classi documentali, per ognuna si stabiliscono i metadati/tag obbligatori e facoltativi)
- I metadati obbligatori per una data classe documentale vengono inseriti nella sezione “mandatory” dell’XML
- I metadati facoltativi per una data classe documentale vengono inseriti nella sezione “extrainfo” dell’XML
- Per entrambe le sezioni sono previste due tipologie “strutturali” di metadati:
 - singletadata, struttura per la memorizzazione di un singolo metadato, ogni “singletadata” è costituito dalla seguente struttura (namespace=dominio delle informazioni, name=nome del metadato, value=valore assunto dal metadato):

```
<singletadata>
  <namespace>conservazione.doc</namespace>
  <name>dataDocumentoTributario</name>
  <value>2015-01-20</value>
</singletadata>
```

- complexmetadata, struttura che permette l’aggregazione di più “singletadata” afferenti ad un comune dominio (es. cap, città, via e civico sono singletadata raggruppabili in un “complexmetadata” denominato “indirizzo”), anche il complexmetadata possiede un proprio namespace e nome:

```
<complexmetadata namespace="conservazione.doc" name="destinanatario"
  namespaceNode="conservazione.soggetti" nodeName="soggetto">
  <singletadata>
  .....
  </singletadata>
  .....
  <singletadata>
  .....
  </singletadata>
<complexmetadata>
```

- DocFly permette di recuperare (esibire) un singolo file o l’indero PdA (la cartella di PdV) solo a partire da un identificativo assegnato da DocFly stesso al termine della messa in conservazione
- La messa in conservazione non è sincrona all’invio in conservazione effettuato da TAKo. L’invio a DocFly si conclude perciò non con la reale messa in conservazione, ma con una semplice presa in carico da parte di DocFly stesso. Per ottenere l’esito della messa in conservazione (e gli ID assegnati da DocFly, utili per una futura esibizione) è necessario prevedere una ulteriore chiamata asincrona verso DocFly da effettuare in tempi successivi all’invio del PdV.

Sulla base di quanto emerso si potrebbe prevedere di:

- Mappare l'unità documentaria sul singolo PdV, ogni PdV di aruba rappresenterebbe così una singola unità documentaria e ogni singolo “file” del PdV rappresenterebbe il singolo componente e sottocomponente previsto dalle specifiche SIP TAKo
- Creare lato DocFLy un PdA per ogni PdV in considerazione che non è possibile recuperare (esibire) il singolo PdV, ma solo l'intera cartella PdA (che potenzialmente potrebbe contenere più PdV); è da valutare questa soluzione in quanto sembra sconsigliata da Aruba (da comprendere per quale motivazione)
- Realizzare un sistema di configurazione specifico per il connettore “DocFly” necessario per:
 - catalogare le “classi documentarie” previste in DocFLy associando ognuna di esse alla corretta “tipologia di unità documentaria” prevista da TAKo
 - definire per ogni tipologia “classe documentaria” i metadati obbligatori e quelli facoltativi previsti
 - mappare ciascun metadato “lato aruba” associandolo al corrispondente metadato previsto dallo standard SIP TAKo
 - memorizzare i parametri specifici della connessione (username, password, URL sistema DocFLy, etc)
- Prevedere nel modello dati generale di TAKo la memorizzazione degli ID assegnati da parte dell'ente conservatore e necessari alla successiva esibizione/download
- Prevedere nell'architettura generale di TAKo, e in relazione alla struttura dei connettori/plugin, che lo stato di messa in conservazione possa essere ottenuto, per alcuni conservatori, solo mediante chiamate asincrone rispetto all'invio in conservazione; questo necessita l'implementazione in TAKo di automatismi/crontab.

Nella progettazione del sistema di configurazione per il connettore aruba, (per la mappatura dei metadati), una volta assunto che un PdV aruba coincide con una unità documentaria TAKo, è necessario tenere particolarmente in considerazione:

- ogni PdV appartiene ad una classe documentaria, è possibile perciò associare con una relazione 1-1 la classe documentaria di Aruba con la relativa tipologia di unità documentaria prevista da TAKo avendo entrambe le medesime finalità
- il PdV di aruba si focalizza fortemente sul file, l'XML del PdV non è altro che un elenco di file, l'unica associazione realizzabile fra PdV e unità documentaria è associare a ciascun componente o sottocomponente (di un documento appartenente all'u.documentaria) un elemento “file” (del PdV aruba). Come diretta conseguenza il sistema di configurazione del connettore per aruba, per quanto riguarda la mappatura verso il PdV, deve avere come punto di vista il singolo file/componente. All'interno del PdV aruba sarà sempre presente almeno un elemento “file” in quanto, per le specifiche SIP TAKo, è sempre obbligatorio il “documento principale” ed almeno un suo componente.

Riepiloghiamo le caratteristiche dell'unità documentaria prevista da TAKo/Parer, utile per comprendere quanto seguirà:

1. *Unità documentaria:*

- A. l'unità documentaria possiede dei metadati "fissi" previsti dallo standard
- B. l'unità documentaria viene classificata per mezzo di una "tipologia unità documentaria" che stabilisce:
 - a. quali siano i metadati specifici per quella unità documentaria
 - b. quali tipologie di documento possano essere presenti nell'unità documentaria e per quale elemento (documento principale, annesso,) la data tipologia di documento possa essere usata
- C. l'unità documentaria è costituita da uno o più documenti (elementi):
 - a. un documento principale (obbligatorio)
 - b. zero o più allegati
 - c. zero o più annessi
 - d. zero o più annotazioni

2. *Documento:*

- A. il documento possiede dei metadati "fissi" previsti dallo standard
- B. il documento viene classificato per mezzo di una "tipologia documento" che stabilisce:
 - a. quali siano i metadati specifici per quella tipologia di documento
 - b. quali componenti e relativi sotto-componenti (di ciascun componente) possano costituire il documento
- C. ogni documento può essere composto da uno o più componenti, ogni componente identifica un file (relativo al documento) che deve essere inviato in conservazione
- D. ogni documento deve avere almeno un componente di tipo (tipo componente) "contenuto"

3. *Componente:*

- A. il componente possiede dei metadati "fissi" previsti dallo standard
- B. il componente viene classificato per mezzo di una "tipologia componente" che stabilisce:
 - a. quali siano i metadati specifici per quella tipologia di componente
- C. Ogni componente può essere costituito da uno o più sotto-componenti, in funzione della tipologia di documento

4. *Sotto-componente:*

- A. Valgono le medesime considerazioni fatte per il componente, non è però obbligatorio che vi sia almeno un sotto-componente di tipo "contenuto" (per ogni "componente" di appartenenza)
- B. Non sono previsti ulteriori sotto-livelli (sotto-sotto-componenti non sono previsti)

Nel seguito l'albero (sintetizzato) delle relazioni fra unità documentaria, documenti e componenti è:

Unità Documentaria

- Documento principale
 - Componente1
 - Sottocomponente1
 - Sottocomponente2
 -
 - Sottocomponente-n
 - Componente2
 -
 - Componente-n
- Allegati
 - Allegato1
 - Componente1
 - Sottocomponente1
 - Sottocomponente2
 -
 - Sottocomponente-n
 - Componente2
 -
 - Componente-n
 - Allegato2
 -
 - Allegato-n
- Annessi
 - Annesso1
 - Componente1
 - Sottocomponente1
 - Sottocomponente2
 -
 - Sottocomponente-n
 - Componente2
 -
 - Componente-n
 - Annesso2
 -
 - Annesso-n
- Annotazioni
 - Annotazione1
 - Componente1
 - Sottocomponente1
 - Sottocomponente2
 -
 - Sottocomponente-n
 - Componente2
 -
 - Componente-n
 - Annotazione2
 -
 - Annotazione-n

Sostanzialmente, stabilita la tipologia di unità documentaria di interesse, come un evento in cascata, si ottiene l'intero indice dei tag che possono presentarsi all'interno dei un XML rappresentante una unità documentaria (per quella tipologia):

1. i “metadati specifici” per quella unità documentaria”
2. quale documenti e relative tipologie di documenti possono essere presenti in:
 - Documento principale
 - Allegati
 - Annessi
 - Annotazioni
3. per ciascuna tipologia di documento (appartenente ad una degli elementi del punto 2)
 - i metadati specifici del documento
 - quali componenti/sottocomponenti possono esservi e i relativi metadati specifici

Quanto segue è un possibile flusso operativo guidato nella configurazione della mappatura per il connettore Aruba:

1. Noto l'elenco delle classi documentarie previste da Aruba (in DocFly), censirle creando in TAKo tante nuove “classe documentaria aruba” quante previste su DocFly. Associare ciascuna “classe documentaria aruba” appena creata alla rispettiva “tipologia di unità documentaria” prevista da TAKo nel suo modello generale. A questo punto diviene implicita (considerando quanto già indicato in precedenza) la seguente associazione: “classe documentaria aruba => “tipologia di unità documentaria” => elenco e struttura dei metadati prevista dal SIP TAKo (per quella specifica tipologia di unità documentaria)”
2. Seleziono, fra quelle create seguendo il punto 1, la “classe documentaria aruba” della quale voglio descrivere la struttura del suo PdV, seleziono per esempio “Fattura attiva”
3. Il fulcro del PdV di aruba è il singolo file, per questo motivo è necessario che, da qui in avanti, il processo di configurazione sia centrato sull'elemento “file”. Nel modello TAKo/Parer il corrispondente del “file aruba” è il “componente” o il “sotto-componente”. Per ogni possibile componente e sotto-componente previsto per la specifica tipologia documentaria scelta (al punto 2) sarà necessario procedere alla creazione di un elemento “file aruba”. Si viene a creare così una associazione 1-1 fra l'elemento “file aruba” e l'elemento “componente o sotto-componente” dell'unità documentaria di TAKo. Le seguenti operazioni vanno quindi ripetute per tutti i componenti eo/ sotto-componenti per la tipologia documentaria in fase di configurazione:
 - A. Creo un nuovo elemento di tipo “file aruba” (ipotizzo di voler mappare il foglio di trasformazione XSLT della fattura elettronica, esso è il sottocomponente (di tipo “Foglio di trasformazione”) del

componente Tipo=“Contenuto” appartenente al documento principale quando esso è tipo documento=“Fattura”

- B. Seleziono il tipo di “elemento” (documento principale, allegato, annesso o annotazione) al quale il componente/sotto-componente si riferisce, scelgo per esempio “Documento principale”
- C. Il sistema mi presenta le tipologie di documento che si possono presentare per il “Documento principale” per questa tipologia documentaria, (es. Fattura, Acconto, Nota di credito, Parcella, ...), scelgo quindi “Fattura”
- D. Il sistema mi presenta le tipologie di componenti possibili per il tipo di documento scelto al punto precedente, nel nostro caso d’esempio è possibile solo il tipo “Contenuto”, lo scelgo
- E. Il sistema mi presenta, se previste, le tipologie di sotto-componente previste per il componente scelto al punto precedente, nel nostro caso d’esempio seleziono “Foglio di trasformazione”
- F. A questo punto il sistema mi permette, grazie all’indice completo dei metadati reso disponibile da l’XMLSipIndex, di selezionare un qualsiasi metadato appartenente al componente (o sotto-componente) oppure all’unità documentaria (fissi o specifici dell’unità documentaria)
- G. Come noto, Aruba prevede due sezioni di metadati, “mandatory” e “extrainfo”, l’interfaccia di configurazione potrebbe essere suddivisa nelle due sezioni per facilitarne l’usabilità:
 - mandatory
 - Aruba prevede due tipologie di metadati (“singlemetadata” o “complexmetadata”), seleziono la tipologia desiderata , le seguenti operazioni vanno ripetute per ciascun metadato previsto nel PdV:
 - se singlemetadata:
 - dichiaro il “datatype aruba”, il connettore deve essere dotato con delle funzioni di cast/conversione, indico così quale funzione usare, ad esempio il formato data usato in aruba potrebbe essere diverso da quello usato in TAKo, oppure un metadato che in TAKo è un decimale, potrebbe essere invece un intero su Aruba
 - namespace, è il dominio di appartenenza del metadato, dipende da aruba
 - name, è il nome da assegnare al metadato nel PdV aruba, dipende da aruba
 - value mapper, seleziono a quale metadato di TAKo associare questo metadato aruba, sono previsti due sottocampi ulteriori:
 - tipo, in una prima fase sarà disponibile solo “unità documentaria” (potrebbe risultare utile per accedere ad altre sorgenti di dati? Esempio un LDAP? Da valutare)
 - id, se tipo=unità documentaria, verrà presentato l’intero indice dei metadati possibili per questa unità documentaria, è l’indice offerto dall’oggetto XMLSipIndex creatosi al punto F; selezionare quindi un qualsiasi metadato appartenente al componente (o sotto-componente) oppure all’unità

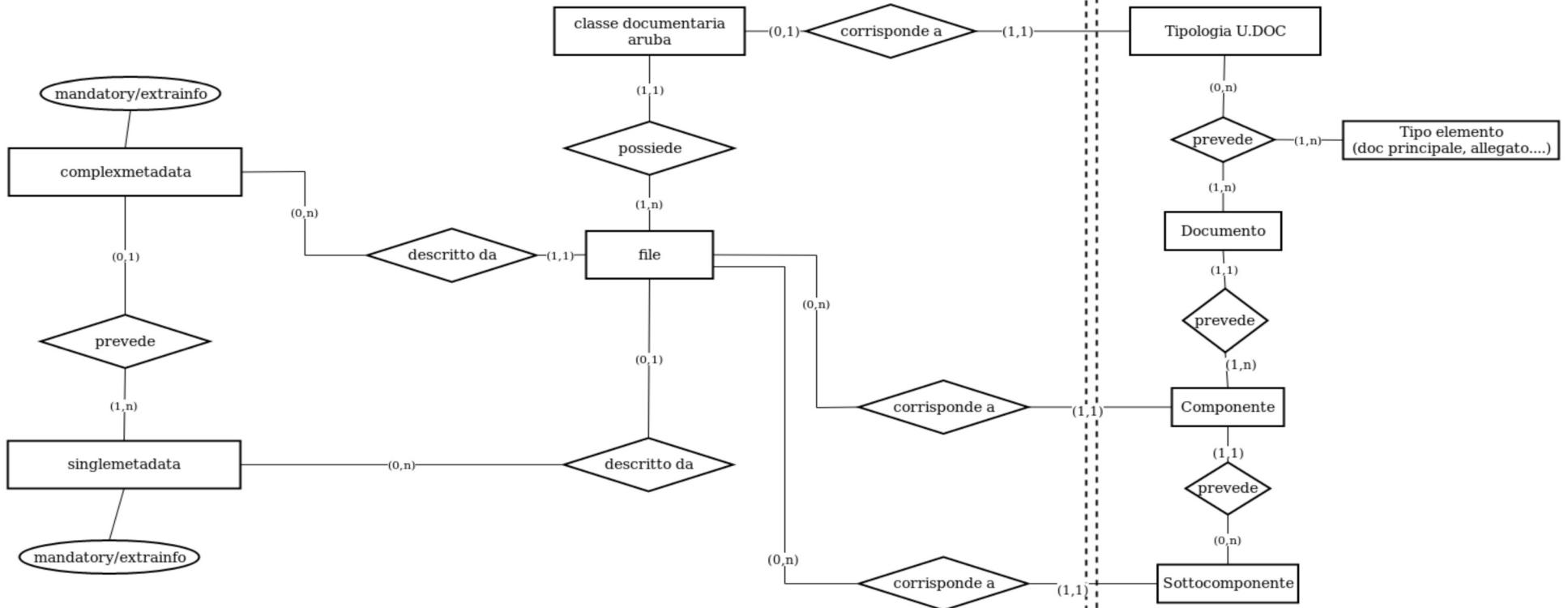
documentaria (fissi o specifici dell'unità documentaria) (i metadati dovrebbero essere identificati col loro XPATH, vedi figura 10 riquadri verdi)

- se complexmetadata:
 - creo una nuova sezione “complexmetadata” indicando:
 - nameSpace, dipende da aruba
 - name, è il nome da assegnare al metadato nel PdV aruba, dipende da aruba
 - namespaceNode, dipende da aruba
 - nodeName, dipende da aruba
 - eseguo, per ogni “singlemetadata” appartenente al “complexmetadata”, quanto indicato nella sezione dei “singlemetadata”
 - extrainfo => ripetere quanto già indicato per la sezione mandatory

4. a questo punto la mappatura risulta completa

Il flusso sopra descritto si propone più come flusso logico, dovrà essere adeguatamente ingegnerizzata l'interfaccia di configurazione per facilitarne le operazioni previste.

Lo schema E/R che segue rappresenta una modellazione di massima, non esaustiva e da rivedere in fase progettuale (anche il modello E/R di TAKo è da progettare definitivamente), che potrebbe essere adottata per il modello dati incaricato di gestire le configurazioni di mappatura del connettore Aruba.



Modello mappatura su connettore Aruba

Figure 10

Catalogo struttura SIP/Unità documentaria

Prendiamo ora in esame le fasi relativi all'invio del PdV ad Aruba, Aruba prevede due operazioni distinte (asincrone) per l'invio in conservazione:

1. invio del PdV a DocFly
2. interrogazione dello stato di conservazione

Questo è dovuto al fatto che, come già indicato in precedenza, l'operazione reale di messa in conservazione viene effettuata da DocFly in tempi successivi all'invio fisico del PdV a DocFly. La risposta data da DocFly al termine dell'operazione di invio del PdV è perciò solamente l'esito di una "presa in carico per elaborazione successiva".

Sono perciò da prevedere, in particolare per il connettore di Aruba, ma più in generale come comportamento generico di ogni connettore sviluppato per TAKo, due operazioni che TAKo dovrà temporizzare/mettere a crontab:

- invio del PdV al conservatore
- interrogazione dello stato di conservazione del PdV inviato in precedenza

1) Operazioni di invio del PdV:

- riceve come input l'oggetto "UnitDocObject" (che rappresenta il pacchetto SIP originale, metadati + URI interni di TAKo per accedere ai file da inviare in conservazione)
- individua la tipologia di unità documentaria
- nota la tipologia, accede al proprio sistema di configurazione e ottiene le configurazioni di mappatura
- seguendo le configurazioni di mappatura procede a produrre il PdV
- invocherà la WS prevista da DocFly per l'invio in conservazione
- attenderà l'esito della chiamata gestendo l'eventuale codice d'errore
- a questo punto il flusso si interrompe, essendo l'esito della messa in conservazione asincrono rispetto a questa chiamata

2) Operazioni per l'interrogazione dello stato di conservazione del PdV:

- interroga lo stato di conservazione del PdV
- se è stato versato, aggiorna lo stato e invia al core di TAKo gli ID necessari per un eventuale procedura di esibizione (download dei file versati)

Questo comportamento "asincrono" rappresenta quindi un vincolo progettuale per TAKo, l'invio sincrono diviene quindi un mero caso "particolare". Anche la restituzione di un ID, da parte del conservatore e necessario per eventuali operazioni di esibizione, restituito nella seconda chiamata asincrona è un forte vincolo progettuale per TAKo. Al momento l'unico sistema sincrono e che non prevede un ID generato dal conservatore è Parer, tutti gli altri conservatori risultano essere asincroni, prevedere un ID oppure entrambi assieme.