Testing e simulazione di servizi erogati in un'architettura SOA: SIL-VIO

A. Pappalardo

Abstract

Con l'avvento di Internet e con la decentralizzazione di ruoli e responsabilità ha preso piede un nuovo modello per lo sviluppo e la cooperazione del software: l'architettura SOA. L'OASIS (Organizzazione per lo sviluppo di standard sull'informazione strutturata) definisce la SOA come "*un paradigma per l'organizzazione e l'utilizzazione delle risorse distribuite che possono essere sotto il controllo di domini di proprietà differenti. Fornisce un mezzo uniforme per offrire, scoprire, interagire ed usare le capacità di produrre gli effetti voluti consistentemente con presupposti e aspettative misurabili*". Per far si che le applicazioni esistenti interagiscano in una architettura SOA si rende necessaria una loro parziale riscrittura per adattarle ai nuovi modelli di comunicazione e di scambio di messaggi oltre che per uniformarle ad una interfaccia comune. SIL-VIO, discusso in questa tesina, è un recente progetto dell'Università degli Studi della Basilicata nato nell'ambito di una convenzione con la Regione Basilicata che verte sul progetto nazionale ICAR. SIL-VIO si pone come obiettivo quello di testare e/o simulare servizi erogati in un'architettura SOA, permettendone una rapida implementazione senza la necessità di adattare immediatamente le applicazioni esistenti. Nel corso di questa tesina introdurremo in primis l'architettura SOA, in seguito inquadreremo il progetto nazionale SPCoop/ICAR, infine sarà discusso nel dettaglio il progetto SIL-VIO.

1. Introduzione architetture SOA

Con la locuzione inglese di **Service-Oriented Architecture** (SOA) [1] si indica un'architettura software atta a supportare l'uso di servizi Web per soddisfare le richieste degli utenti così da consentire l'utilizzo delle singole applicazioni come *componenti* di un più complesso processo di business.

Architettura significa insieme di regole finalizzate a raggiungere un determinato obiettivo e quindi anche una SOA si avvale di una serie di strumenti che permettono di definire e di descrivere i flussi informativi affinché siano leggibili via Web. Da un punto di vista logico, per strutturare i servizi servono:

- Un linguaggio comune che permetta ai clienti di individuare, richiedere e ottenere i servizi senza ambiguità o errori;
- Regole chiare, persistenti, con cui redigere contratti tra fornitori di servizi e clienti, regole che devono essere comprensibili alla componente umana e gestibili dalla componente informatica;
- Un canale di comunicazione sicuro, semplice, ad alta prestazione (Enterprise Service Bus - ESB), su cui possano avvenire la comunicazione tra fornitori e clienti e i necessari colloqui di servizio.

L'ESB è l'elemento fondamentale dell'architettura Soa in quanto agisce come layer middleware di astrazione all'interno dei sistemi informativi e rende disponibili i servizi tra i vari sistemi e le nuove applicazioni.

Da un punto di vista funzionale, la coesistenza e il funzionamento dell'insieme dei servizi si basano proprio sull'esistenza di un'infrastruttura tecnica unica di comunicazione (il Bus). I servizi sono imperniati in prevalenza sull'adozione di tecnologie distribuite basate su Xml e una volta resi pubblici attraverso il Bus devono essere disponibili per qualsiasi utente abbia accesso al Bus. C'è infine un'evoluzione del concetto di interfaccia: da Api (Application Programming Interface) a vero e proprio contratto, che comprende tutta una serie di informazioni quali struttura, semantica, vincoli, sicurezza, versioni, tecnologie di mapping.

SOA definisce, quindi, una modalità di descrivere i componenti (servizi) con caratteristiche orientate al riutilizzo e all'integrazione; SOA è uno stile architetturale basato sul concetto di servizio, che rappresenta quindi l'elemento strutturale su cui le applicazioni vengono sviluppate. Un servizio è, in prima analisi, una funzionalità di business realizzata tramite un componente che rispetta un'interfaccia, la quale deve essere pubblicabile sulla rete, ricercabile e invocabile in maniera indipendente dal linguaggio e dalla piattaforma.

I paradigmi SOA sono stati corroborati dalle tecnologie dei Web-Services (SOAP, WSDL, UDDI), le quali si adattano perfettamente a questo nuovo modello architetturale coinvolgendone tutti i soggetti principali (Fig. 1) :

- Service Consumer: l'entità che richiede il servizio; può essere un modulo di un'applicazione o un altro servizio.
- Service Provider: l'entità che fornisce il servizio e che ne espone l'interfaccia.
- Service Contract: definisce il formato per la richiesta di un servizio e della relativa risposta.
- Service Registry: registro in rete dei servizi consultabili.

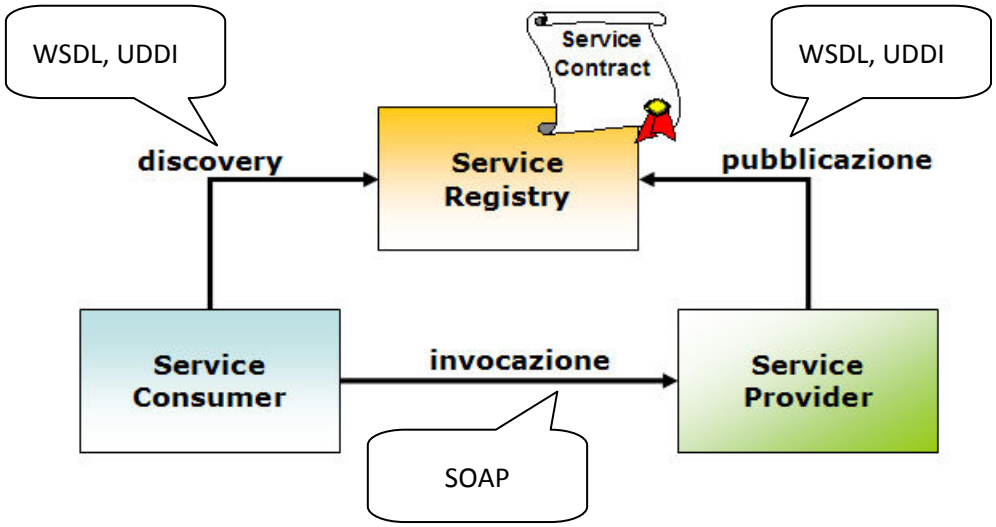


Figura 1: i ruoli SOA

2. SPCoop

Sulla base delle specifiche architetturali SOA nel 2005 nasce, ad opera del CNIPA [4], il progetto nazionale SPCoop ("Sistema Pubblico di Cooperazione") che si pone come obiettivo la standardizzazione dei meccanismi di cooperazione applicativa tra le pubbliche amministrazioni italiane, definendo un framework per l'interoperabilità fra servizi applicativi. In particolare, SPCoop introduce gli elementi tecnologici che comporranno l'infrastruttura di cooperazione tra le pubbliche amministrazioni italiane, come la *porta di dominio*, la *busta e-Gov*, l'*accordo di servizio*, il *registro dei servizi*. Di fatto, SPCoop può essere definito come l'*Enterprise* Service Bus (ESB) della pubblica amministrazione italiana.

In un classico scenario di cooperazione applicativa proposto da SPCoop, due Sistemi Informativi Locali (SIL) appartenenti a reti private diverse, possono comunicare tra di loro instaurando una comunicazione punto-punto tra le proprie porte di dominio. La porta di dominio [2] rappresenta il

gateway attraverso cui ogni amministrazione offre e consuma servizi. E' possibile pensare che sia il punto di ingresso e di uscita unico per il dominio dell'amministrazione, come illustrato in Fig. 2.

La porta di dominio presenta una duplice interfaccia: una interna per comunicare con i servizi applicativi ed una esterna per scambiare messaggi con altre porte di dominio. I messaggi scambiati con i SIL del proprio dominio sono messaggi applicativi e possono essere basati su protocolli e standard diversi. Viceversa, i messaggi che una porta di dominio scambia con altre porte di dominio devono rispettare un formato ed un protocollo standard. Questi messaggi, detti messaggi SPCoop, devono essere messaggi basati sul formato SOAP 1.1 with attachments, utilizzare il protocollo HTTPS, e prevedere una serie di intestazioni standard che vanno sotto il nome di busta di e-gov.

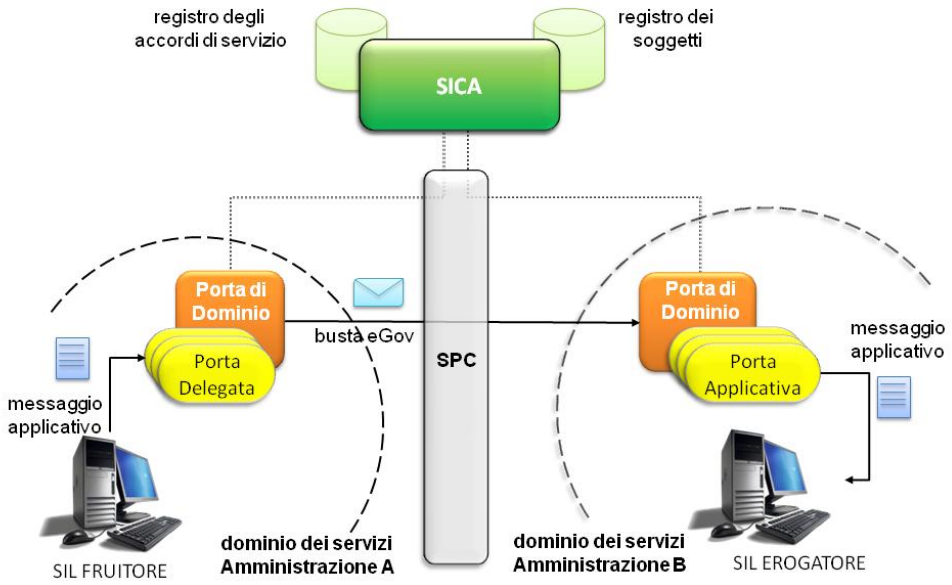


Figura 2: Architettura SPCoop

L'ultimo elemento dell'architettura SPCoop sono i *Servizi Infrastrutturali per la Cooperazione* Applicativa (SICA). Compito dei SICA è mantenere opportuni registri che vengono interrogati dalle porte di dominio per l'elaborazione dei messaggi applicativi. In particolare, i servizi SICA contengono:

- un registro dei soggetti abilitati alla comunicazione nell'ambito del sistema;
- un registro degli accordi di servizio che prescrivono le modalità e i requisiti per lo svolgimento corretto della cooperazione applicativa.

In sintesi, un accordo di servizio (Service Contract) è un documento – in formato XML – che regola l'erogazione di un servizio applicativo da parte di un sistema locale ai fruitori che vogliono accedervi, descrivendo in dettaglio:

- i soggetti coinvolti (l'erogatore e i fruitori autorizzati ad accedere ai servizi);
- i servizi che sono oggetto dell'accordo, con le relative interfacce (sostanzialmente, frammenti di codice WSDL)
- i profili di collaborazione secondo cui i servizi vengono erogati (sincrono, asincrono, one-way);
- le politiche di sicurezza e la qualità del servizio prescritta.

E' previsto che le porte di dominio contattino il registro centrale per ottenere informazioni sui soggetti e sugli accordi di servizio.

2.1. SPCoop – Esempio di comunicazione

Analizziamo ora i passi della comunicazione tra due SIL appartenenti a due domini diversi:

- per cominciare, il SIL fruitore contatta la propria porta di dominio, chiedendo di inoltrare al sistema erogatore del servizio un messaggio applicativo; il componente della porta di dominio demandato a ricevere messaggi applicativi di richiesta da parte dei SIL si chiama *porta delegata*;
- la porta delegata effettua il processo di imbustamento del messaggio applicativo ricevuto, per trasformarlo in una busta e-gov ed instradarla sul bus; in particolare, reperisce l'indirizzo della porta del dominio a cui appartiene il SIL erogatore del servizio richiesto, e inoltra la busta;
- il componente della porta di dominio che riceve messagi e-gov di richiesta si chiama *porta applicativa*; ha il compito di effettuare lo sbustamento della richiesta ricevuta, ed inoltrarla al SIL erogatore;
- Il messaggio di risposta seguirà un percorso simile

2.2. SPCoop e ICAR

ICAR [6] è un progetto nazionale diretto dal CISIS [5] che coinvolge 16 regioni ed una provincia autonoma. L'obiettivo è la creazione di un'infrastruttura concreta di cooperazione tra le amministrazioni secondo le specifiche fornite dal CNIPA.

Il progetto ICAR prevede, oltre alla realizzazione dell'infrastruttura di base necessaria alla cooperazione applicativa, lo sviluppo di sette task applicativi; sette casi di studio in specifici domini applicativi della cooperazione applicativa interregionale. Essi hanno l'obiettivo di sperimentare e dimostrare l'efficacia dei servizi infrastrutturali di interoperabilità e cooperazione applicativa, realizzati con gli interventi infrastrutturali di base, in alcuni scenari applicativi di livello interregionale.

A tal fine, sono state previste le attività di analisi dei requisiti, il progetto e la realizzazione delle interfacce tra le applicazioni esistenti a livello regionale con l'Infrastruttura ed i servizi di base per la Interoperabilità e Cooperazione Applicativa, che permettono l'attivazione di servizi di cooperazione applicativa interregionale in specifici domini applicativi, considerati significativi e di prioritario interesse per lo sviluppo operativo della Community Network interregionale.

Il progetto Icar prevede che la cooperazione avvenga seguendo i paradigmi dettati dall'Enterprise Application Integration, una disciplina, emersa recentemente, che si pone come obiettivo di integrare in modo rapido e performante applicazioni enterprise per mezzo di interfacce applicative standard; sono, quindi, le singole applicazioni partecipanti alla cooperazione a dover essere modificate ed estese, integrandosi con i Web Services.

3. SIL-VIO

SIL-VIO [7], Sistema Informativo Locale Virtuale per l'InterOperabilità, è un recente progetto dell'Università degli Studi della Basilicata nato nell'ambito di una convenzione con la Regione Basilicata che verte sul progetto nazionale SPCoop/ICAR. SIL-VIO si pone come obiettivo quello di testare e/o simulare servizi erogati in un'architettura SOA basata su Enterprise Service Bus (ESB), permettendone una rapida implementazione senza la necessità di adattare immediatamente le applicazioni esistenti. Data un'interfaccia applicativa (wsdl o accordo di servizio nel caso specifico del dominio di cooperazione ICAR) SIL-VIO è in grado di implementare ed esporre automaticamente i servizi necessari dal lato dell'erogatore e di invocarli correttamente dal lato fruitore.

3.1. Architettura di SIL-VIO

SIL-VIO è una applicazione web (Fig. 3), basata su tecnologie open source Java, che dovrà esporre i servizi descritti dai wsdl inseriti dall'utente, per mezzo del framework Apache CXF, e prevedere l'interazione con varie sorgenti di dati.

Figura 3: Interfaccia SIL-VIO

Nello scenario più generale (Fig. 4) di cooperazione tra architetture SOA differenti, SIL-VIO si pone fra l'ESB e il database del sistema informativo che usufruisce (o eroga) il servizio. La comunicazione verso l'esterno avviene tramite messaggi SOAP mentre i servizi esposti necessitano di un'interazione con i DB locali, a cui il sistema si interfaccia mediante la tecnologia JPA (Java Persistence API).

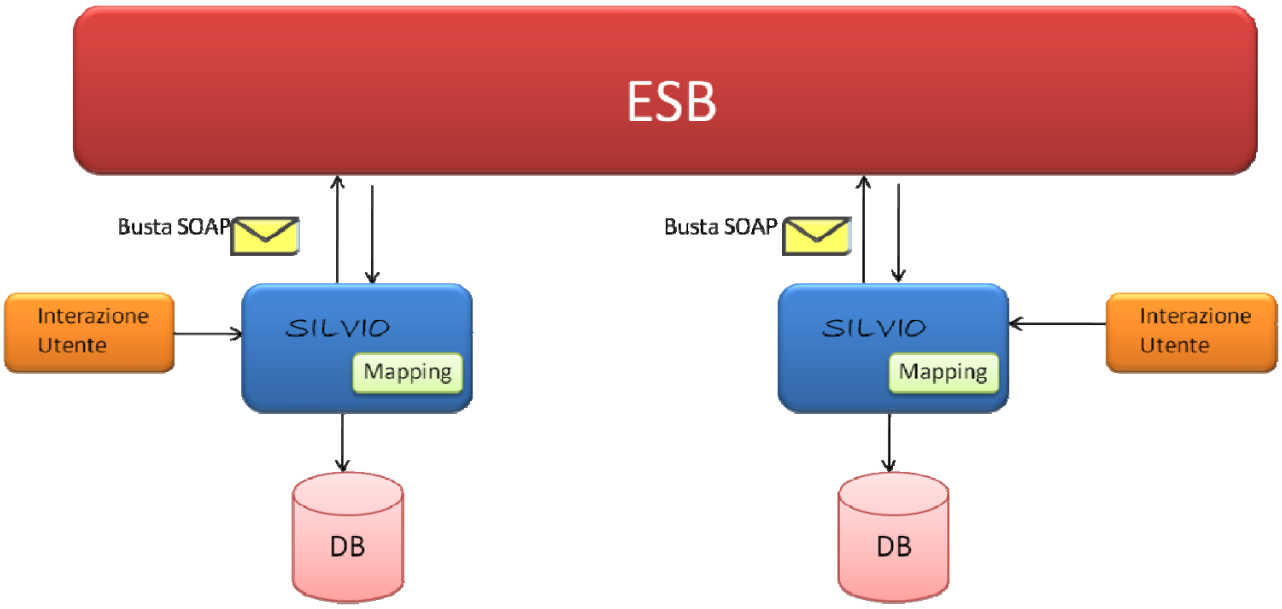


Figura 4: Architettura SIL-VIO

4. Tecnologia e Configurazione

Per far si che SIL-VIO possa erogare o fruire un particolare servizio specificato in un wsdl o in un Accordo di Servizio è necessario provvedere ad una fase preliminare di configurazione in cui, oltre a dettagli sistemistici, l'utente dovrà fornire essenzialmente tre informazioni:

1. Da dove e come prelevare i dati necessari alla creazione del messaggio SOAP
2. Come creare il messaggio di richiesta/risposta partendo dai dati in possesso
3. Come elaborare e salvare all'interno di un database i dati contenuti in un messaggio SOAP in ingresso

Sarà quindi compito dell'utente fornire tutte quelle informazioni atte a ricreare il flusso di trasformazioni (Fig. 5) necessario a SIL-VIO per replicare correttamente il funzionamento del servizio in oggetto.

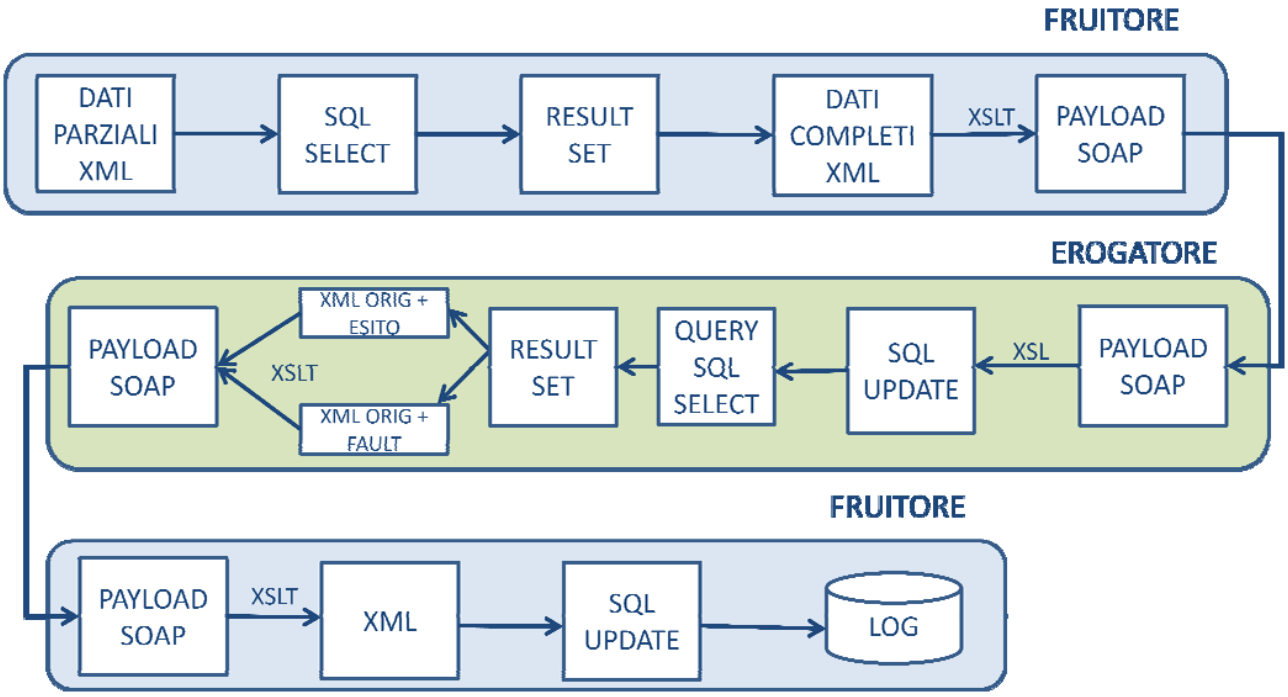


Figura 5: Flusso trasformazioni SIL-VIO

Nel seguito esemplificheremo l'uso delle tecnologie alla base della configurazione di SIL-VIO facendo riferimento alla fruizione del servizio IdentificazioneAssistito del task applicativo AP1 [8] (Cooperazioni e Compensazioni Sanitarie Interregionali) del progetto ICAR; tale servizio permette l'acquisizione dei dati anagrafici di un utente del sistema sanitario e del suo medico di base partendo dal Codice Fiscale o dal codice SSR (Servizio Sanitario Regionale) .

4.1. Prelevare i dati

In alcuni casi è necessario inserire nel messaggio SOAP dati prelevati da un database e/o richiesti interattivamente all'utente e/o costanti. Per estrarre i dati da un Db dovranno essere forniti a SIL-VIO i parametri per l'accesso e le query SQL; quest'ultime potranno essere anche parametrizzate ed in tal caso il sistema chiederà interattivamente all'utente di completarle ad ogni esecuzione.

```
SELECT * FROM tabella WHERE tabella.campo = ?
```

Esempio concreto di IdentificazioneAssistito del task AP1.

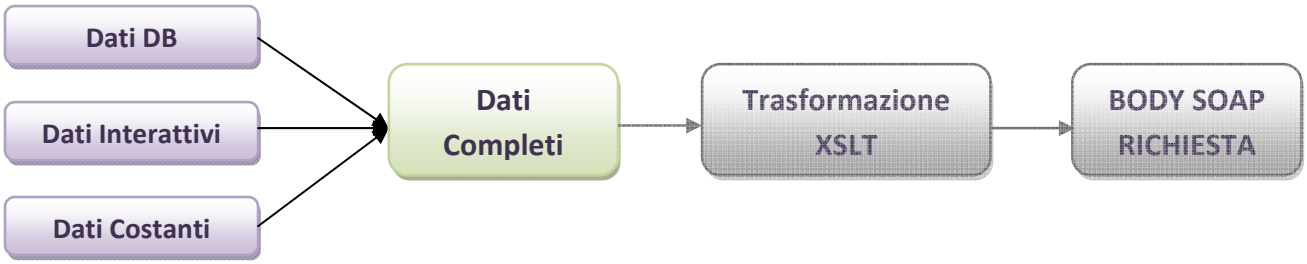
Campi della richiesta	Valori
IdentificazioneRegioneMittente	170
…	…
IdentificazioneRegioneDestinatario	100
…	…
CodiceFiscaleAssistito	AAAAAA00A00A000A
ImportoTicket	100,00

La regione mittente è un dato che non varia ad ogni esecuzione dell'istanza → ***Dato Costante***

La regione del destinatario è un dato che varia ad ogni esecuzione dell'istanza → ***Dato Interattivo***

I dati del ricoverato sono prelevati dalla base di dati → ***Dato da DB***

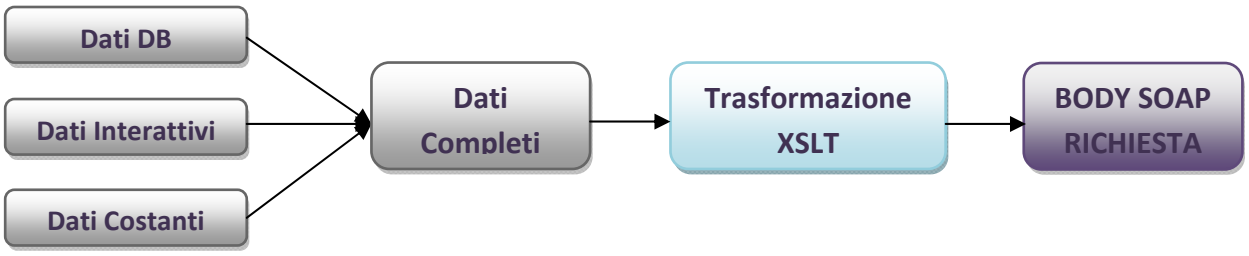
Una volta estratti tutti i dati dalle varie sorgenti, SIL-VIO si occuperà di raggrupparli in un unico documento XML.



4.2. Creazione messaggio SOAP

Completata la definizione di tutti i dati necessari per generare il messaggio di richiesta, occorre specificare come questi andranno a comporre il body del messaggio SOAP.

Per fare ciò bisogna definire un file XSLT [9] che permetta di trasformare il documento XML, contenente i dati completi, generato da SIL-VIO, nella richiesta da inviare all'erogatore. Il sistema offre supporto alla creazione del file XSLT fornendo lo schema XSD del documento contenente i dati completi (schema XSD sorgente) e lo schema XSD del messaggio che dovrà essere inviato (schema XSD destinazione). Per generare il file XSD della sorgente a partire dai dati da estrarre dal db, il sistema effettua preventivamente la query fornita dall'utente (escludendo la clausola WHERE); dal risultato viene ricavata la lista dei campi, nel formato tabella.campo, che può essere aggiunta al documento XML dei dati completi. Nel caso in cui la query non dovesse produrre alcun risultato, il sistema non sarà in grado di generare in maniera corretta il file XSD della sorgente. Un volta che l'utente avrà a disposizione i due schemi XSD, potrà realizzare manualmente le trasformazioni XSLT oppure avvalersi dell'ausilio di strumenti automatici sia Open Source (Jamper[1]) che commerciali (Altova MapForce® [2]) che ne facilitano enormemente il compito. Usando tali strumenti, infatti, l'utente dovrà limitarsi a individuare dei mapping tra elementi della sorgente ed elementi della destinazione e sarà il sistema a generare il file XSLT corretto.



A questo punto il sistema possiede tutti gli elementi per poter creare il messaggio body della richiesta. Di seguito ne è riportato un esempio.

```
<richiesta_RichiestaRispostaAsincrona_ IdentificazioneAssistito >
   <Intestazione>
      <InfoMittente>
            <Regione>170</Regione>
      </InfoMittente>
      <InfoDestinatario>
            <Regione>100</Regione>
      </InfoDestinatario>
      <DataOra>2001-12-17T09:30:47.0Z</DataOra>
      <IdMessaggio>0</IdMessaggio>
      <IdComunicazione>0</IdComunicazione>
   </Intestazione>
   <IdentificazioneAssistito>
      <DatiAnagrafici>
            <CodiceFiscale>AAAAAA00A00A000A</CodiceFiscale>
            <Sesso>1</Sesso>
            <ComuneResidenza>000000</ComuneResidenza>
      </DatiAnagrafici>
      <DatiPrestazioniAmbulatoriali>
            <DatiRicetta>
                  <Data>2008-06-13</Data>
                  <PosizioneUtenteTicket>1</PosizioneUtenteTicket>
                  <ImportoTicket>100,00</ImportoTicket>
                  <ImportoTotale>100,00</ImportoTotale>
            </DatiRicetta>
      </DatiPrestazioniAmbulatoriali>
   </IdentificazioneAssistito >
</richiesta_RichiestaRispostaAsincrona_SegnalazioneFruizionePrestazioni>
```

La regione mittente è un dato che non varia ad ogni esecuzione dell'istanza → ***Dato Costante***

La regione del destinatario è un dato che varia ad ogni esecuzione dell'istanza → ***Dato Variabile***

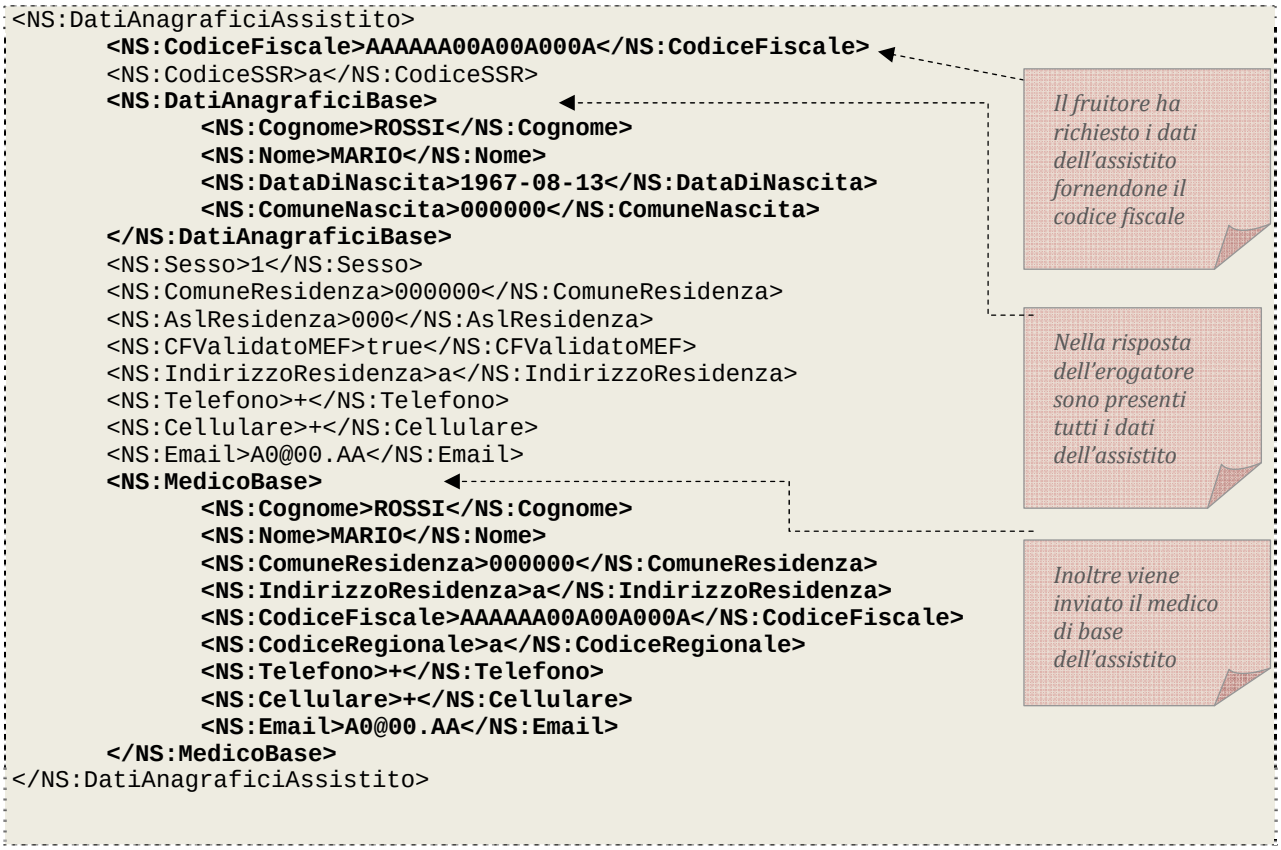
I dati del ricoverato sono prelevati dalla base di dati → ***Dato da DB***

[1] http://jamper.sourceforge.net/
[2] http://www.altova.com/MapForce

4.3. Elaborazione messaggio SOAP

Una volta inviato il messaggio SOAP all'erogatore, il fruitore resta in attesa della risposta che potrà contenere informazioni da dover salvare in una propria base di dati. Per effettuare questa operazione il sistema necessita di un file di trasformazione XSLT che permetta di estrarre i dati dalla risposta dell'erogatore e al contempo inserirli in una base di dati per mezzo di una query SQL. Attualmente, al meglio delle nostre conoscenze, non esiste nessuno strumento né proprietario né open source che possa offrire una qualsivoglia forma di supporto nella stesura di questo tipo di trasformazioni che quindi dovranno necessariamente essere scritte manualmente dall'utente.

Esempio di risposta da parte dell'erogatore :



Esempio di file XSLT che permette l'elaborazione della risposta dell'erogatore da parte del fruitore che inserirà i dati del paziente in un proprio database:

```
<xsl:template match="/">
      INSERT INTO paziente (nome, cognome, codice_fiscale, data_nascita, sesso,
comune_residenza, codice_ssr) VALUES(
            <!-- elemento radice -->
               '<xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_Identificazi
oneAssistito/DatiAnagraficiAssistito/DatiAnagraficiBase/Nome"/>',
                  '<xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_Identificazi
oneAssistito/DatiAnagraficiAssistito/DatiAnagraficiBase/Cognome"/>',
                  '<xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_Identificazi
oneAssistito/DatiAnagraficiAssistito/CodiceFiscale"/>',
                  '<xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_Identificazi
oneAssistito/DatiAnagraficiAssistito/DatiAnagraficiBase/DataDiNascita"/>',
                  '<xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_Identificazi
oneAssistito/DatiAnagraficiAssistito/Sesso"/>',
                  '<xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_Identificazi
oneAssistito/DatiAnagraficiAssistito/ComuneResidenza"/>',
                  '<xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_Identificazi
oneAssistito/DatiAnagraficiAssistito/CodiceSSR"/>'
                  );
      </xsl:template>
```

L'utente deve quindi fornire la query per l'inserimento dei valori nella base di dati. Si noti come per estrarre i dati dal messaggio SOAP si faccia largo uso della sintassi XPATH [10] attraverso la quale viene prelevato il valore di un elemento specificando la sua posizione all'intero dell'albero che rappresenta l'intero documento.

In alcuni casi, è possibile che dalla richiesta arrivino informazioni che dovranno essere salvate in tabelle differenti dello stesso database e che dovranno essere correlate tra loro attraverso un identificatore (nel nostro esempio vi è la correlazione medico-paziente attraverso chiave esterna). La sintassi XSLT non permette di specificare in alcun modo questo tipo di correlazione, né è possibile, per l'utente, scrivere precedentemente query corrette poiché non può conoscere il valore delle keys attribuite alle ennuple che solitamente sono generate dal dbms stesso.

ID	Cognome	Nome	Codice_Fiscale
01	Rossi	Mario	Rssmra56945poil
02	Verdi	Chiara	Vrdchr89034jilk
03	Bianchi	Enrico	Bncnrc78oagaet
…	…		

Tabella Medico

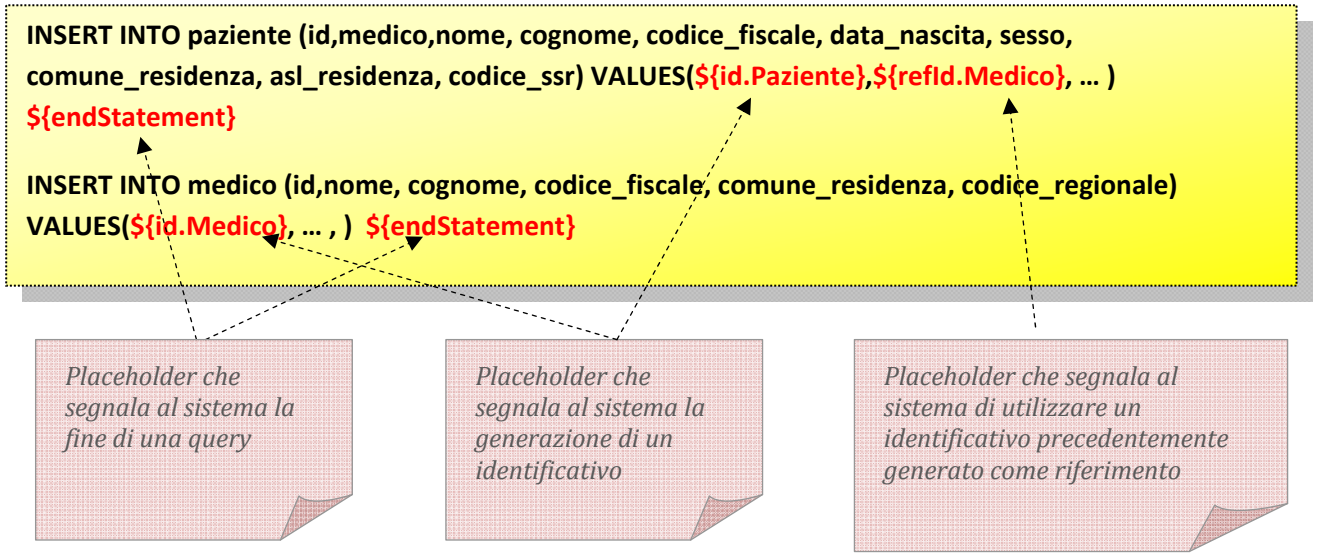
mID	Cognome	Nome	Medico_id
01	Celeste	Rossella	01
02	Colli	Andrea	02
03	Felice	Rocco	01
…	…		

Tabella Paziente

4.3.1 Placeholder

Per superare il problema delle INSERT correlate il sistema offre la possibilità di utilizzare nel file di trasformazione alcuni placeholder che per mezzo di una semplice sintassi permettono di specificare le informazioni aggiuntive necessarie per l'elaborazione. Il placeholder ${id.tabella} indicherà a SIL-VIO di dover gestire automaticamente l'attribuzione degli ID alle ennuple da inserire; si farà invece uso del placeholder ${refId.tabella} per correlare l'ennupla in oggetto ad un'altra ennupla appartenente ad una tabella differente.

Per generare l'ID da attribuire alla INSERT fornita dall'utente, il sistema per prima cosa effettua una SELECT sulla tabella in oggetto calcolando il MAX degli ID presenti; successivamente incrementa questo valore di 1, lo attribuisce alla query fornita e la esegue. Se la query non va a buon fine, il sistema presume che siano stati violati alcuni vincoli di chiave; per ovviare a ciò, SIL-VIO effettua una SELECT utilizzando tutti i campi a disposizione per verificare la reale presenza della ennupla nella tabella e ne estrae l'ID. Se la selezione restituisce una ennupla vuota o un risultato errato sarà sollevata un'eccezione. In caso di esito positivo, l'ID estratto viene salvato e sostituito al posto degli eventuali placeholder ${refId.tabella} che ad esso facevano riferimento.



5. SIL-VIO: utenti e concorrenti

SIL-VIO è uno strumento concepito per essere il più duttile possibile e come tale si rivolge a diverse tipologie di utenti, i quali potranno usarlo per raggiungere differenti obbiettivi. Da una parte, gli sviluppatori dell'infrastruttura di base alla cooperazione hanno la necessità di testare il corretto funzionamento dell'architettura e di valutarne la scalabilità, a tale scopo SIL-VIO mette a disposizione una modalità di esecuzione che permette l'invio e la gestione di messaggi a payload costante e l'esecuzione sequenziale e/o concorrente di un numero arbitrario di istanze. Dall'altra parte, gli sviluppatori dei wsdl o, nell'ottica ICAR, degli Accordi di servizio, potranno avvantaggiarsi delle caratteristiche di flessibilità e di facile riconfigurazione di SIL-VIO per testare velocemente ogni piccola modifica ai servizi senza dover ogni volta riscrivere i web-services e riadattare i sistemi. Sempre nell'ottica ICAR, i partecipanti alla sperimentazione dei task applicativi, potranno immediatamente iniziare a testare quanto prodotto senza la necessità di nessun sistema reale su cui appoggiarsi. Infine, in alcune situazioni, SIL-VIO potrebbe essere usato direttamente in produzione sovrapponendosi ai sistemi legacy ed evitandone così la riscrittura.

Sul mercato esistono vari prodotti commerciali che si pongono come obbiettivo la simulazione di servizi (lato erogatore) a partire dai rispettivi WSDL, ma, al meglio delle nostre conoscenze, nessuno offre una configurazione sofisticata come quella di SIL-VIO che permetta non solo di simulare ma bensì di replicare il funzionamento reale che dovrebbe presentare il servizio.

Un esempio di prodotto commerciale è Crosscheck Networks SOAPSimulator® [3] che si pone come obiettivo, dichiarato dagli sviluppatori, quello di realizzare un ambiente di facile utilizzo per la simulazione di servizi al fine di rendere possibile la scrittura di test su quest'ultimi ancor prima che vengano effettivamente implementati. Il punto di partenza per l'utilizzo di questo strumento è sempre un file wsdl, che il sistema usa per generare un albero di tutte le operation simulabili; l'utente potrà decidere quali operation implementare configurandole individualmente. Al contrario di SIL-VIO in cui l'utente configura genericamente, per ogni input corretto, il comportamento di un servizio, SoapSimulator prevede che l'utente fornisca per ogni singolo input, che ci si aspetta, un output corrispondente. È evidente come siano diverse le filosofie alla base dei due sistemi: mentre SIL-VIO vuole replicare il comportamento reale di un servizio, SoapSimulator crea un dummy service che non farà altro che rispondere a input predefiniti con output predefiniti, senza tra l'altro interagire con nessuna sorgente di dati. E' doveroso sottolineare come SoapSimulator si occupi anche di altri aspetti legati ai servizi web, ed alle architetture SOA in generale, come la definizione di Best Practices e dei criteri per la loro valutazione o come la misura dell'aderenza agli standard di sicurezza.

[3] http://www.crosschecknet.com/

6. Conclusioni e sviluppi futuri

SIL-VIO, Sistema Informativo Locale Virtuale per l'InterOperabilità, è un recente progetto dell'Università degli Studi della Basilicata che si pone come obiettivo quello di testare e/o simulare servizi erogati in un'architettura SOA basata su Enterprise Service Bus (ESB), permettendone una rapida implementazione a partire dalle loro interfacce.

Il sistema, che si presenta come una applicazione Web ed è basato su tecnologie open source del mondo Java, permette una configurazione accurata del comportamento del servizio simulato oltre che del suo modo di interagire con l'utente e con un database, riuscendo a replicare il funzionamento che presenterebbe un web-service reale realizzato ad hoc. Punto cruciale del suo approccio al problema è l'uso della tecnologia XSLT che consente di realizzare tutta la catena di trasformazioni (Fig. 5) necessarie alla corretta implementazione del servizio.

In una prossima fase di sviluppo si provvederà a rendere SIL-VIO in grado di interagire con web-services preesistenti da usare come ulteriori possibili sorgenti di dati ed ancora si cercherà di attribuire al sistema capacità di composizione e coreografia di web-services.

Bibliografia

[1] Wikipedia – Service-oriented architecture. http://it.wikipedia.org/wiki/Service-oriented_architecture

[2] Wikipedia - Porta di Dominio. http://it.wikipedia.org/wiki/Porta_di_dominio.

[3] G. Mecca, A. Pappalardo, S. Raunich, il Gruppo di Sviluppo ICAR. “Soluzioni Infrastrutturali Open Source per il Sistema Pubblico di Cooperazione Applicativa”. Symposium on Advanced Database Systems (SEBD 2008), Mondello (PA) - Italy, 22-25 june 2008.

[4] Centro Nazionale per l'Informatica nella Pubblica Amministrazione. Requisiti e specifiche funzionali del SPCoop. http://www.cnipa.gov.it/site/itit/In_primo_piano/Sistema_Pubblico_di_Connettività_(SPC)/Servizi_di_interoperabilità_evoluta_e_cooperazione_applicativa/. 2005

[5] CISIS - Centro Interregionale per i Sistemi informatici, Geografici e Statistici. http://www.cisis.it/.

[6] Progetto Nazionale ICAR (“Interoperabilità e Cooperazione Applicativa tra le Regioni). http://www.progettoicar.it/Home.aspx

[7] SIL-VIO - Sistema Informativo Locale Virtuale per l'InterOperabilità. http://freesbee.unibas.it/index.php.

[8] Task Applicativo AP1 – Progetto ICAR. http://www.progettoicar.it/ViewCategory.aspx?catid=a64edf851c664da2a69417464773a5f9 .

[9] XSL Transformations (XSLT) - W3C Recommendation. http://www.w3.org/TR/xslt.html .

[10] XML Path Language (XPath) - W3C Recommendation. http://www.w3.org/TR/xpath.