

# Aspetti di Interoperabilità della Specifica SPCoop nell'implementazione OpenSPCoop 1.0

Andrea Corradini, Tito Flagella, Andrea Poli

Dipartimento di Informatica, Università di Pisa (<http://www.di.unipi.it>)

[corradini@di.unipi.it](mailto:corradini@di.unipi.it)

Link.it (<http://www.link.it>)

[{tflagella, apoli}@link.it](mailto:{tflagella, apoli}@link.it)

## Abstract

Avviato nel 2004 con un'approfondita fase di analisi e progettazione svolta tramite due tesi di Laurea Specialistiche dell'Università di Pisa, il progetto OpenSPCoop sta oggi per rilasciare la versione 1.0 del software, che rappresenta un'implementazione completa della specifica SPCoop, sia per quanto attiene ai componenti periferici (Porte di Dominio), che ai componenti centrali (Registro dei Servizi e Gestore Eventi). In questo articolo, dopo aver fatto il punto del lavoro progettuale e implementativo realizzato per la versione 1.0 di OpenSPCoop, vengono discussi una serie di aspetti relativi all'interoperabilità tra diverse implementazioni della specifica SPCoop, proponendo alcune possibili disambiguazioni.

## Introduzione

OpenSPCoop (<http://openspcoop.org>) è un progetto finalizzato alla realizzazione di un insieme di componenti Open Source aderenti alle specifiche per la *Cooperazione Applicativa* nella Pubblica Amministrazione, recentemente rilasciate dal Centro Nazionale per l'Informatica nella Pubblica Amministrazione (CNIPA) e note con il nome di Servizio Pubblico di Cooperazione (SPCoop). Come per ogni nuova specifica non banale, anche per SPCoop è di fondamentale importanza l'esistenza di un'implementazione di riferimento open source, che permetta di sperimentare in maniera condivisa i vari concetti proposti nella specifica, evidenziando e proponendo possibili soluzioni per le potenziali ambiguità o debolezze della specifica stessa. Il progetto OpenSPCoop nasce sostanzialmente con questo obiettivo, enfatizzando i noti vantaggi dell'approccio open source per indirizzare i seguenti aspetti:

- *Interoperabilità*, OpenSPCoop intende rappresentare un riferimento per disambiguare diverse possibili interpretazioni della specifica SPCoop;
- *Sicurezza*, l'apertura del codice assicura quelle caratteristiche di trasparenza del codice ormai

considerate un atto dovuto in molti settori della sicurezza informatica;

- *Comunità d'Utenza*, OpenSPCoop tende a fungere da catalizzatore per le esperienze e le competenze degli utenti, permettendo di ricapitalizzarle in risultati concreti e riusabili;
- *Innovazione*, un'implementazione Open Source è il veicolo ideale per proporre delle implementazioni condivisibili di quanto non ancora trattato nelle specifiche SPCoop.

Preceduta da un'approfondita fase di analisi e di progettazione svolta presso il Dipartimento di Informatica dell'Università di Pisa, la prima release del software OpenSPCoop è stata rilasciata da Link.it il 27 ottobre del 2005. Questa prima release è stata poi seguita da frequenti nuovi rilasci, anche grazie al feedback e al contributo dei primi utenti del software. Attorno al sito del progetto si è sviluppata sin dal primo momento una comunità di utenti e sviluppatori molto qualificati, provenienti da aziende italiane, pubbliche amministrazioni locali e centrali e centri di ricerca. L'interesse diffuso per il progetto ha dimostrato tra l'altro la bontà della scelta di una soluzione open source in un settore così critico come quello indirizzato dalla specifica SPCoop.

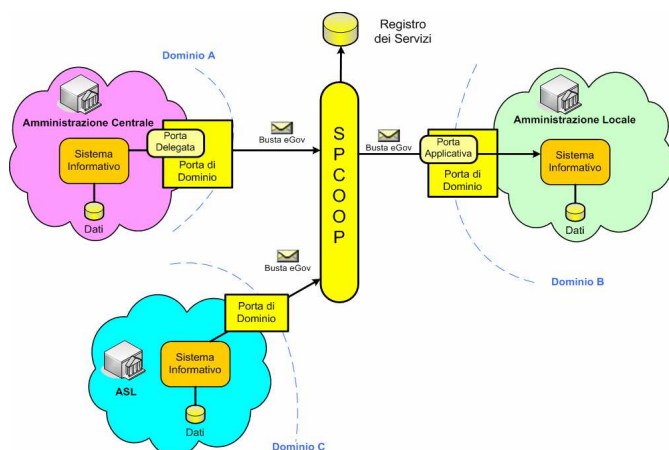
## La Specifica SPCoop

Prima che si cominciasse ad affermare l'idea della standardizzazione di un paradigma di cooperazione applicativa, le comunicazioni nella P.A. avvenivano tramite collegamenti punto-punto tra i server applicativi interessati. Poiché tipicamente questi server erano dislocati su reti private, raggiungibili quindi esclusivamente da altri server dislocati sulla loro stessa rete privata, era necessario realizzare reti virtuali private (VPN) tra le amministrazioni interessate a cooperare.

Questa soluzione si è rapidamente dimostrata inadatta nella gestione del processo di eGovernment, che prevede che due qualunque enti debbano essere potenzialmente in grado di comunicare tra loro. SPCoop risolve questo problema imponendo un'infrastruttura standard di comunicazione tra le amministrazioni pubbliche. In tal modo, una volta che

l'infrastruttura sia diventata completamente operativa, sarà sufficiente il collegamento di un'amministrazione all'infrastruttura SPCoop per abilitarla alla comunicazione con qualunque altra amministrazione italiana ed europea.

La figura seguente mostra i principali componenti parte dell'infrastruttura SPCoop, in particolare: la busta eGov, la Porta di Dominio, le porte delegate e applicative, il Registro dei Servizi e il Gestore Eventi.



Uno dei componenti principali della specifica SPCoop è la Porta di Dominio (PdD), che delimita il confine di responsabilità di un ente o soggetto amministrativo e racchiude al suo interno tutte le applicazioni da esso gestite. Le comunicazioni da e verso un dominio devono quindi attraversare la sua Porta di Dominio (PdD).

Le Porte di Dominio si parlano tra di loro scambiandosi richieste e risposte in un formato standard, denominato busta eGov, che è sostanzialmente una specializzazione di un messaggio SOAP, esteso con un apposito header per definire le caratteristiche del protocollo SPCoop. Poiché il formato della busta non è parlato nativamente (e non è previsto che lo sia) dalle applicazioni, la Porta di Dominio deve anche occuparsi di convertire le richieste applicative in formato proprietario nel formato busta eGov.

La porta delegata e la porta applicativa costituiscono gli elementi della Porta di Dominio che mediano gli accessi tra i sistemi interni agli enti e l'infrastruttura SPCoop e sono quindi spesso riferiti come componenti di integrazione. In particolare la Porta Delegata è utilizzata come proxy per l'accesso al servizio destinazione, mentre la Porta Applicativa deve essere in grado di gestire la consegna dei contenuti applicativi a un server interno al dominio destinazione.

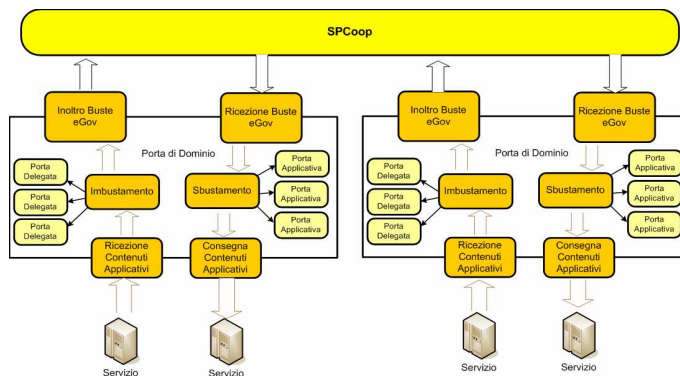
La specifica SPCoop prevede poi la presenza di un registro dei soggetti e degli accordi di servizio tra essi stipulati. Si prevede l'esistenza di un registro di primo livello, gestito dal CNIPA e che includa tutti i servizi ufficiali SPCoop a livello nazionale, e di registri di

secondo livello che possano contenere un sottoinsieme dei servizi SPCoop.

La specifica prevede infine la presenza di un Gestore Eventi per permettere lo scambio di buste eGov secondo l'architettura EDA.

## La Porta di Dominio in OpenSPCoop

Uno degli aspetti più innovativi del progetto OpenSPCoop è la soluzione adottata per la realizzazione dei componenti di integrazione (porte delegate e applicative) della Porta di Dominio. L'abilitazione di nuove porte delegate e/o applicative non richiede infatti la programmazione di componenti applicativi ad-hoc, come normalmente richiesto da altre implementazioni, ma è basata invece su servizi generici utilizzabili da qualunque applicazione. Questi servizi consultano dei repository xml che descrivono le specifiche porte delegate e applicative residenti sulla Porta di Dominio per decidere come gestire le richieste in arrivo. In funzione di queste descrizioni e dello stato delle transazioni eGov in corso, le richieste ricevute dalla Porta di Dominio vengono quindi recapitate al destinatario o girate alla porta di Dominio competente. La soluzione, illustrata nella figura successiva, ha il grande vantaggio di non richiedere la realizzazione di uno specifico componente di integrazione per ogni servizio da utilizzare (porta delegata) o da fornire (porta applicativa), riducendo quindi drasticamente la complessità di attivazione di nuovi servizi SPCoop.



Utilizzando questo approccio si evita anche che sulla porta di dominio venga eseguito codice applicativo sviluppato ad-hoc per i singoli servizi, eliminando così i forti rischi, dal punto di vista dell'affidabilità e della sicurezza, legati all'esecuzione di codice di terze parti su un componente così critico dell'infrastruttura SPCoop.

## L'interazione tra i Servizi Applicativi e la Porta di Dominio

Differentemente da quanto avviene per il componente di cooperazione (busta eGov), per il componente di integrazione la specifica SPCoop si limita a presentare un esempio di massima di una sua possibile realizzazione. In OpenSPCoop è stato quindi necessario progettare completamente le modalità per

l'interazione delle porte delegate e applicative della Porta di Dominio e le applicazioni interne al Dominio dell'Ente (i Servizi Applicativi).

Le modalità di integrazione previste in OpenSPCoop sono due:

1. modalità trasparente: prevede che il servizio applicativo utilizzi (in caso di porta delegata) o esponga (in caso di porta applicativa) le interfacce applicative native dei servizi, esattamente come registrate negli accordi di servizio; in tal caso la Porta di Dominio agisce come un proxy trasparente con funzionalità di imbustamento e sbustamento eGov dei messaggi applicativi; utilizzando questa modalità, gli applicativi potranno continuare ad operare esattamente come se stessero interagendo direttamente con il servizio applicativo dell'altro Ente;
2. uso del Servizio "IntegrationManager" della PdD: prevede di utilizzare le interfacce di un web service di Integrazione messo a disposizione dalla Porta di Dominio per la spedizione e/o la ricezione di messaggi applicativi da parte dei servizi applicativi del proprio Dominio di Servizi.

### **Modalità d'uso trasparente dei Servizi**

Questa modalità prevede che il servizio applicativo utilizzi (in caso di porta delegata) o esponga (in caso di porta applicativa) le interfacce applicative native dei servizi, così come registrate negli accordi di servizio; in tal caso la Porta di Dominio agisce come un proxy SOAP trasparente con funzionalità di imbustamento e sbustamento eGov dei messaggi applicativi. Utilizzando questa modalità, gli applicativi potranno quindi continuare ad operare esattamente come se stessero interagendo direttamente con il servizio applicativo dell'altro Ente.

L'invocazione della porta delegata in modalità trasparente può essere realizzata tramite gli strumenti del linguaggio di programmazione nativo del servizio applicativo, utilizzando ad esempio stub creati tramite il proprio ambiente di sviluppo Web Services (ad esempio wsdl2java in Axis), facendo riferimento direttamente al WSDL del servizio erogatore registrato nell'accordo di servizio. In questo caso le principali modifiche rispetto all'invocazione dell'effettivo servizio destinazione saranno:

- la URL utilizzata per l'invocazione http, che dovrà essere quella corrispondente alla porta delegata del servizio esposta dalla PdD, anziché quella dell'effettivo servizio destinazione;
- la gestione degli errori aggiuntivi, che possono essere introdotti dall'interazione con la Porta di Dominio; questi errori sono riconosciuti dal particolare FaultActor "org.openspcoop.org".

Nel riquadro seguente mostriamo un frammento di codice applicativo per l'invocazione di una porta delegata in modalità trasparente, usando Axis 1.4 come web-services engine, che evidenzia in grassetto le

modifiche minime che sono necessarie al codice originale per l'interazione con la Porta di Dominio.

```
try {
    HelloWSServiceLocator locator = new
    HelloWSServiceLocator();

    locator.setHelloWorldEndpointAddress("http://fruit
    ore:8080/openspcoop/PD/HelloWorld");

    HelloWS port = locator.getHelloWorld();
    String msg = port.getDate();

} catch (AxisFault e) {

if("org.openspcoop.pdd".equals(e.getFaultActor()))
{
    System.out.println("Errore PdD");
    System.out.println("Code:
"+e.getFaultCode());
    System.out.println("String:
"+e.getFaultString());
}else{
    System.out.println("SOAPFault Applicativo");
    System.out.println("Actor:
"+e.getFaultActor());
    System.out.println("Code:
"+e.getFaultCode());
    System.out.println("String:
"+e.getFaultString());
}
} catch (Exception e) {
    System.out.println("ClientError:
"+e.getMessage());
    e.printStackTrace();
}
```

Come evidenziato nell'esempio, il particolare codice di eccezione generato dalla Porta di Dominio potrà essere ottenuto tramite il campo FaultCode del messaggio di Fault SOAP.

### **Uso del Servizio IntegrationManager**

In alternativa alla modalità trasparente, è possibile utilizzare il servizio di IntegrationManager, messo a disposizione dalla Porta di Dominio per la spedizione e/o la ricezione di messaggi applicativi da parte dei servizi applicativi del proprio Dominio di Servizi. L'interfaccia WSDL completa dell'Integration manager è consultabile alla URL

<http://openspcoop.org/doc/IntegrationManager.wsdl>.

Per maggiori informazioni sul suo uso si rimanda alla documentazione di OpenSPCoop.

### **Il Registro Servizi di OpenSPCoop**

Il ruolo del Registro dei Servizi è quello di permettere la registrazione e la successiva interrogazione degli Accordi di Servizio (AS) di SPCoop. A partire dagli AS si snodano i vari riferimenti ai soggetti erogatori e fruitori, alle interfacce dei servizi erogati e fruiti, alle politiche di sicurezza, agli SLA, ed in generale a tutto quanto riferisce alle politiche di cooperazione applicativa concordate tra i Soggetti interessati.

OpenSPCoop supporta quattro versioni del registro dei servizi:

- *Registro XML*, il registro è realizzato tramite un singolo file xml. Questa versione è perfettamente funzionale e molto semplice da gestire, adatta quindi a scenari non molto dinamici e con un basso numero di soggetti e servizi in gioco; questo registro può essere gestito solo manualmente tramite l'editing del file xml.
- *Registro DB*, il registro è realizzato come un database relazionale; questo registro è gestibile tramite l'interfaccia web di gestione del Registro.
- *Registro Web*, il registro è realizzato come un sito web che mantiene un insieme di oggetti descritti in xml e accessibili via http; questo registro è gestibile tramite l'interfaccia web di gestione del Registro.
- *Registro UDDI*, il registro è realizzato tramite un registro in tecnologia UDDI che indicizza un insieme di oggetti descritti in xml, mantenuti in un repository accessibile via http; questo registro è gestibile tramite l'interfaccia web di gestione del Registro.

Ognuna di queste implementazioni ha i suoi vantaggi e i suoi svantaggi in termini di flessibilità e di performance e la scelta del Registro o dei Registri da utilizzare dipenderà quindi dalla specificità dei vari contesti in cui saranno utilizzati. Tutte le varie implementazioni condividono comunque il formato di rappresentazione esterna degli Accordi di Servizio. Tale rappresentazione è in effetti un'estensione di OpenSPCoop, perchè la specifica SPCoop non fornisce un formalismo relativo alla descrizione degli AS. Il formato di un accordo di servizio OpenSPCoop consiste in un insieme di file, distribuiti separatamente o spesso raccolti in un archivio compresso secondo la codifica zip, aderenti ai vari formati standard indirizzati dalla specifica SPCoop dell'accordo, come WSDL o WSBL, e da un manifest che ha la funzione di descrivere l'accordo, includendo anche gli opportuni riferimenti agli altri file parte dell'accordo. Se l'accordo è distribuito come archivio, il manifest è incluso nell'archivio con il path META-INF/manifest.xml. Il manifest è espresso in xml, ed è conforme allo schema xml disponibile alla URL <http://openspcoop.org/schemi/registro.xsd>.

## **Il Gestore Eventi di OpenSPCoop**

OpenSPCoop 1.0 include un'applicazione Web Service che realizza lo scenario di coordinamento ad eventi (architettura EDA); si tratta di un'applicazione esterna, del tutto correlata dalla Porta di Dominio, con la quale è integrata tramite l'uso delle porte delegate e applicative.

Se l'invio di un messaggio al gestore Eventi risulta però come una normale invocazione di servizio SPCoop dal Soggetto che pubblica l'evento al Soggetto che fornisce il Servizio di Gestione degli Eventi, la ritrasmissione dello stesso messaggio a tutti i soggetti

interessati richiede invece la spedizione di una diversa copia dello stesso messaggio ad ogni destinatario.

Per gestire nella maniera più efficiente possibile questa situazione, è stata ottimizzata l'integrazione del Gestore Eventi con la Porta di Dominio, tramite l'uso di porte applicative e delegate per riferimento, appositamente realizzate nella versione 1.0 di OpenSPCoop. Usando la modalità "per riferimento", la Porta di Dominio scambia con il Gestore Eventi soltanto l'id eGov del messaggio da spedire e non il suo contenuto che viene mantenuto nella coda dei messaggi in transito sulla Porta di Dominio. In questo modo si evita di far viaggiare più volte lo stesso messaggio tra l'applicazione Gestore Eventi e la porta di dominio che le sta davanti per ogni diversa destinazione.

## **Aspetti di Interoperabilità**

Uno degli obiettivi del progetto OpenSPCoop era quello di favorire la sperimentazione e la discussione su alcuni aspetti della specifica SPCoop, in particolare su quelli che potessero avere un maggiore impatto dal punto di vista dell'interoperabilità tra implementazioni diverse. In questa sezione saranno analizzati alcuni di questi aspetti, la maggior parte dei quali emersi nel corso di vari progetti in cui OpenSPCoop è stato impiegato interoperando con altre implementazioni SPCoop. Per affrontare i problemi elencati nel seguito, l'implementazione di OpenSPCoop permette di configurare alcuni comportamenti della Porta di Dominio a livello delle singole porte delegate e applicative, in modo da poter correttamente interoperare con implementazioni diverse della specifica SPCoop.

### ***L'interoperabilità dei profili Asincroni***

Descriviamo di seguito due possibili comportamenti della PdD in caso di invocazione di una porta delegata abbinata a un servizio asincrono. Un primo comportamento possibile è quello di restituire immediatamente una risposta al Servizio Applicativo mittente. Nel frattempo, in maniera del tutto disaccoppiata dall'interazione con il mittente, si procede all'invio della busta alla PdD del destinatario. Tale PdD replica con una ricevuta con SoapBody vuoto, provvedendo poi, in maniera del tutto disaccoppiata rispetto all'interazione con la PdD mittente, all'invocazione del servizio applicativo destinatario. Un secondo comportamento è invece quello di lasciare il richiedente in attesa fino alla ricezione della ricevuta della prima richiesta asincrona, restituendo quindi al richiedente il soapBody ottenuto nella ricevuta. In questa implementazione, la porta applicativa della Porta destinataria non ignora la risposta del servizio applicativo, ma la inserisce nella ricevuta stessa. La prima interazione asincrona viene quindi implementata in maniera analoga ad una invocazione del profilo Sincrono.

La Porta di Dominio di OpenSPCoop implementa entrambe le modalità descritte, tramite configurazione delle porte delegate ed applicative associate ai servizi asincroni. Le due interpretazioni sollevano però un problema di interoperabilità.

Ad esempio, utilizzando la prima modalità per la PdD destinazione e la seconda per la PdD mittente, la PdD mittente riceverà una ricevuta con un SoapBody vuoto. Il Servizio Applicativo mittente, però, si aspetterà una risposta compliant con il WSDL del servizio e potrebbe rifiutare la risposta ricevuta.

Il problema non si presenterebbe se la specifica per i casi del profilo Oneway, di entrambe le interazioni dell'Asincrono Simmetrico e della richiesta dell'Asincrono Asimmetrico, precisasse:

- che la Porta Delegata risponde al Servizio Applicativo, senza aspettare la ricevuta (e non solo la risposta) della Porta di Dominio destinatario.

oppure

- che il wsdl del servizio applicativo non deve prevedere una risposta (nel wsdl:binding una operation di output non deve essere presente), o comunque l'output non potrà essere recapitato al richiedente.

oppure

- che la specifica definisse un formato di messaggio standard da far restituire dalla PdD Mittente al Servizio Applicativo, come già avviene per il caso di errori applicativi (documento 'Sistema Pubblico di cooperazione: Porta Di Dominio' a pg 41).

oppure

- che i due profili Asincroni (Simmetrico e Asimmetrico) sono da implementare come coppie di scambi sincroni, restituendo quindi al richiedente il reale contenuto della risposta del servizio applicativo destinazione. Questo farebbe però perdere ai profili asincroni gran parte del loro significato, rendendoli inutilizzabili in tutte quelle situazioni in cui non si può o non si vuole dipendere dalla disponibilità di una sincronizzazione tra mittente e destinatario.

### ***Campi opzionali della busta***

Alcuni campi della busta sono opzionali perchè esistono sicuramente tipologie di buste che non li contengono. In alcuni casi però potrebbe essere obbligatorio il loro uso. La specifica dovrebbe chiarire, in funzione di ogni specifica istanza di busta, quali campi sono opzionali. Ad esempio, il Servizio può non essere usato in una busta ad-hoc di riscontro, ma deve essere usato in una richiesta di servizio. Al momento però una busta che non è correlata ad altre buste precedenti, e non include il servizio, pur non avendo alcun senso, è formalmente corretta e deve essere accettata da una Porta di Dominio, pur non avendo alcun effetto collaterale.

### ***Gestione a run time dell'Header e-Gov***

Una serie di scelte sul protocollo SPCoop vengono attualmente fatte dal mittente al momento della generazione della busta. Ad esempio:

- Profilo di Collaborazione
- ConfermaRicezione e inoltro con/senza duplicati
- Consegnata in ordine
- Scadenza
- Indirizzo telematico

E' molto rischioso però che il destinatario assuma queste informazioni dalla busta ricevuta, e sarebbe preferibile che le acquisisse/verificasse dall'accordo di servizio. In OpenSPCoop queste informazioni fanno parte dell'Accordo di Servizio e possono essere consultate a run-time dalle Porte di Dominio.

### ***Ruolo dei Riscontri***

La funzionalità dei riscontri sembra utile solo nel caso del profilo OneWay, mentre può creare una serie di possibili ambiguità per gli altri profili. In particolare:

- per il ProfiloSincrono, la risposta è di fatto un riscontro della richiesta;
- per il ProfiloAsincrono, la ricevuta è di fatto un riscontro della richiesta;

Sarebbe utile che la specifica precisasse che i riscontri possono essere implementati solo su profilo Oneway. Altrimenti, se i riscontri devono effettivamente essere implementati anche per gli altri profili, si pongono i seguenti quesiti:

- Caso Sincrono: la Richiesta e Risposta devono essere entrambe riscontrabili? In tal caso il riscontro della richiesta sarebbe utile fosse obbligatorio restituirlo nella risposta.
- Caso Asincrono:
  - le buste devono essere rispedito se non arriva la ricevuta o se non arriva il riscontro?
  - le ricevute devono essere anch'esse riscontrabili, devono cioè a loro volta possedere l'elemento confermaRicezione che richiede un riscontro?

### ***Gestione delle Risposte***

La specifica è ambigua sul fatto che i messaggi SPCoopErrore, i Riscontri e le Ricevute asincrone vadano inviati su di una nuova connessione o meno (reply http). Sarebbe utile precisare che possono essere utilizzate entrambe le possibilità, o che debbano essere utilizzate:

- la stessa connessione per il profilo sincrono;
- una nuova connessione per i profili oneway e asincrono; in questo caso la nuova connessione permette di utilizzare connessioni asincrone tra le porte in caso di presenza di più porte sul percorso della busta (routing).

### ***Interoperabilità tra PdD e Registri***

Andrebbero definiti i formati dei dati contenuti nei Registri e i protocolli di interazione tra le Porte e i Registri. Questo aspetto non riguarda l'interoperabilità tra le porte ma l'interoperabilità tra Porte di Dominio e Registri SICA.

### ***Riscontro ad hoc***

Una busta contenente un riscontro generato ad hoc è autosufficiente di per sé per la validazione del riscontro (contiene l'ID del riscontro). La specifica non dice però quali altri elementi debba contenere:

- solo i soggetti
- oltre ai soggetti anche:
  - i. servizio e azione della busta che aveva confermaRicezione=true
  - ii. profiloDiCollaborazione della busta che aveva confermaRicezione=true
  - iii. un riferimentoMessaggio con l'ID della busta riscontrata (già presente nel riscontro)

### ***Azione nei profili Asincroni***

Esiste un'ambiguità relativa al campo azione della Risposta/RicevutaRisposta asincrona simmetrica e della RichiestaStato/Risposta asincrona asimmetrica. Il servizio da utilizzare in queste buste è quello indicato come servizio correlato nelle precedenti buste di Richiesta/RicevutaRichiesta.

Andrebbe però precisato quale valore debba assumere il campo azione, se la stessa delle precedenti richieste o nessuna azione.

## **8. Bibliografia**

[R0] Il Progetto OpenSPCoop,  
<http://www.openspcoop.org>

[R1] Andrea Corradini, Tito Flagella e Andrea Poli, "OpenSPCoop: un'Implementazione della Specifica di Cooperazione Applicativa per la Pubblica Amministrazione Italiana", ESPERTA, sessione di OSS2006

[R2] Tesi di Laurea Specialistica in Tecnologie Informatiche - Università di Pisa, OpenSPCoop: un'implementazione della Specifica di Cooperazione Applicativa per la Pubblica Amministrazione Italiana, Febbraio 2006, autore Andrea Poli, relatori Andrea Corradini, Tito Flagella.

[R3] Tesi di Laurea Specialistica in Tecnologie Informatiche - Università di Pisa, Progettazione di un framework Open Source per la cooperazione applicativa nella Pubblica Amministrazione, luglio 2005, autore Ruggero Barsacchi, relatori Andrea Corradini, Tito Flagella.

[R4] SPC, "Sistema pubblico di cooperazione: Architettura, Versione 1.0", CNIPA, 25 Novembre 2004.

[R5] SPC, "Sistema pubblico di cooperazione: Porta di Dominio, Versione 1.0", CNIPA, 14 Ottobre 2005

[R6] SPC, "Specifiche della Busta di e-Government, Edizione 1.0", CNIPA, 21 Aprile 2004

[R7] SPC, "Sistema pubblico di cooperazione: Busta di e-Gov, Versione 1.1", CNIPA, 14 Ottobre 2005

[R8] SPC, "Sistema pubblico di cooperazione: Accordo di Servizio, Versione 1.0", CNIPA, 14 Ottobre 2005

[R9] SPC, "Sistema pubblico di cooperazione: Servizi di Registro, Versione 1.0", CNIPA, 14 Ottobre 2005

[R10] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, D. Winer, "Simple Object Access Protocol (SOAP) 1.1", W3C, 8 Maggio 2000

[R11] J. Barton, S. Thatte, H. F. Nielsen, "SOAP Messages with Attachments", W3C, 11 Dicembre 2000

[R12] E. Christensen, F. Curbera, G. Meredith, S. Weerawarna, "Web Services Description Language (WSDL) 1.1", W3C, 15 Marzo 2001