

# OpenSPCoop: un Progetto Open Source per la Cooperazione Applicativa nella Pubblica Amministrazione

**Andrea Corradini**

Dipartimento di Informatica, Pisa

andrea@di.unipi.it

<http://www.di.unipi.it>

**Tito Flagella**

Link.it

tflagella@link.it

<http://www.link.it>

## Abstract

Avviato nel 2004 con un'approfondita fase di analisi e progettazione svolta tramite due tesi di Laurea Specialistica 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 dapprima inquadrano la specifica SPCoop nel contesto tecnologico dell'integrazione applicativa in ambito enterprise basata su web services. Quindi mostriamo che la soluzione architettonica proposta da OpenSPCoop per la componente di integrazione della Porta di Dominio è innovativa e allo stato dell'arte. Infine, dopo aver brevemente descritto gli altri componenti fondamentali del progetto, discutiamo alcune problematiche relative alla proponibilità di soluzioni basate su software open source nel caso di bandi di fornitura di software custom per le Pubbliche Amministrazioni.

## 1. Introduzione

OpenSPCoop (<http://openspcoop.org>, [1]) 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) [2-7]. 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 [8,9], la prima release del software OpenSPCoop è stata rilasciata da Link.it il 27 ottobre 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. I circa 1500 download del software dalla creazione del progetto ad oggi e i circa 230 iscritti alle liste del progetto testimoniano l'interesse diffuso per il progetto; questo 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.

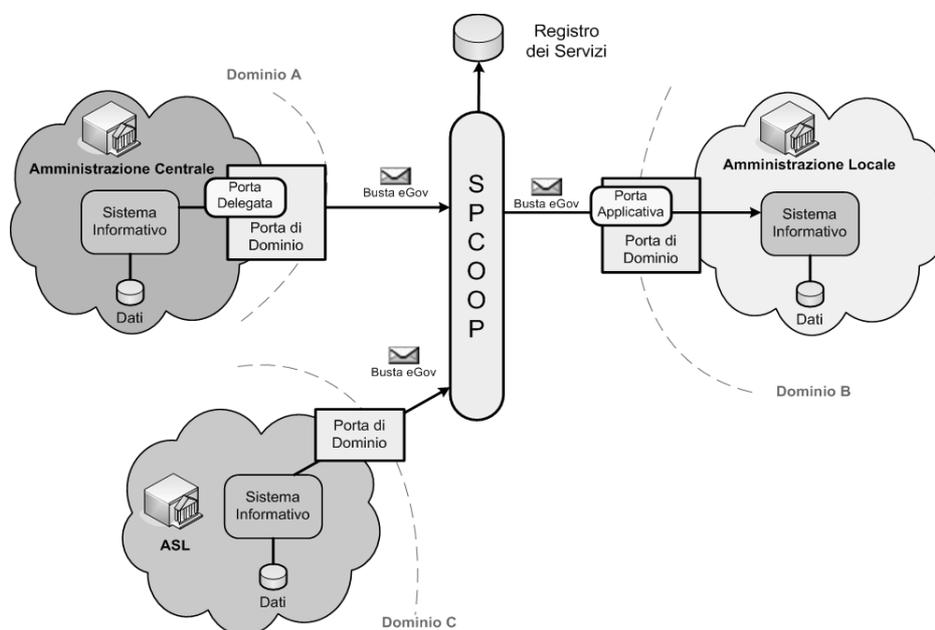
In questo articolo dapprima inquadrano la specifica SPCoop nel contesto tecnologico dell'integrazione applicativa in ambito enterprise basata su web services. Quindi mostriamo che la soluzione architettonica proposta da OpenSPCoop per la componente di integrazione della Porta di Dominio è innovativa e allo stato dell'arte. Infine, dopo aver brevemente descritto gli altri componenti fondamentali del progetto, discutiamo alcune problematiche relative alla proponibilità di soluzioni open source nel caso di bandi di fornitura di software custom per le P.A..

## 2. 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 [2-7] 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 dell'infrastruttura SPCoop: la busta eGov, la Porta di Dominio, le porte delegate e applicative, e il Registro dei Servizi.



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. Le PdD 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.

La porta delegata e la porta applicativa costituiscono gli elementi della PdD che mediano gli accessi tra i sistemi interni agli enti e l'infrastruttura SPCoop. 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, che includa tutti i servizi ufficiali SPCoop a livello nazionale, e di registri di secondo livello che possano contenere un sottoinsieme di tali servizi. La specifica prevede infine la presenza di un Gestore Eventi per permettere lo scambio di buste eGov secondo l'architettura *event-driven* (EDA).

### 3. La Specifica SPCoop nel Contesto dell'Architettura Web Services

Dal punto di vista del contesto tecnologico, la specifica SPCoop è fortemente basata sugli standard dei Web Services. In particolare, essa richiede la compatibilità dei messaggi applicativi con lo standard SOAP 1.1 [10], in accordo alla specifica WS-I Basic Profile 1.1 [11]. La specifica SPCoop comprende due aspetti principali:

1. il formato di un header di protocollo SOAP, comunemente riferito come busta eGov, che permette la gestione di numerosi aspetti del protocollo di comunicazione tra Enti Pubblici, quali l'indirizzamento, l'affidabilità della consegna e la gestione dei duplicati;<sup>1</sup>
2. un insieme di norme organizzative, che identificano alcuni componenti (principalmente le Porte di Dominio e i Registri di Servizi) e il loro ruolo di supporto alla cooperazione tra i servizi applicativi dei singoli domini. In particolare la Porta di Dominio ha un ruolo di mediazione tra i servizi applicativi interni al dominio e le Porte di Dominio degli altri Enti.

Poichè non è previsto che il formato della busta eGov sia parlato nativamente dalle applicazioni, la Porta di Dominio deve anche occuparsi di convertire le richieste applicative in formato proprietario nel formato di busta eGov. Facendo riferimento a questa problematica, i compiti della Porta di Dominio vengono solitamente divisi in due diverse componenti: il componente di integrazione e quello di cooperazione. Mentre la specifica SPCoop copre completamente gli aspetti relativi al componente di cooperazione, per quanto attiene al componente di integrazione essa si limita a presentare un esempio di massima di una possibile realizzazione. Per questo motivo, i vari prodotti finora realizzati si differenziano significativamente proprio per quanto attiene al componente di integrazione, cioè a come costruire le buste a partire dai dati forniti dall'applicativo.

Il problema rientra nella problematica più generale dell'uso dei Web Services per l'Integrazione Applicativa in ambito Enterprise (EAI), problema di importanza centrale negli ambienti Enterprise, e per il quale si sono andate evolvendo soluzioni sempre più sofisticate [12].

#### ***Approccio "End to End"***

Il problema è stato gestito per molto tempo, e in larga parte ancora adesso, direttamente dagli applicativi (mittente e richiedente), che si occupavano in prima persona di gestire sia le modalità di integrazione (imbustamento dei contenuti applicativi in SOAP) che quelle di cooperazione (gestione

---

<sup>1</sup> A questo proposito bisogna notare che quasi tutti questi aspetti sono oggetto di specifici standard dell'architettura Web Services, come WS-Addressing e WS-Reliability. La versione 1.1 della specifica della busta eGov [5] è però arrivata prima del consolidamento di questi standard; comunque viene esplicitamente previsto un progressivo adeguamento nel tempo agli standard consolidati.

dei vari servizi a livello di header SOAP). Queste soluzioni sono state realizzate tramite Web Services Framework del tipo di Axis, Axis2 o XFire.<sup>2</sup>

### ***Approccio “Enterprise Service Bus”***

In vista di un significativo incremento dei servizi realizzati tramite tecnologia Web Services all'interno delle organizzazioni, sono stati sviluppati ambienti che forniscono servizi infrastrutturali comuni per la gestione dell'affidabilità, il routing e la trasformazione dei servizi web. Tali soluzioni, note come Enterprise Service Bus (ESB), forniscono quindi servizi comuni sia di Integrazione che di Cooperazione. Tra gli ESB più noti ci sono Mule, ServiceMix, Celtix e Artix.<sup>3</sup>

Gli ESB hanno costituito un significativo passo avanti per la gestione delle problematiche tipiche dell'EAI. Tuttavia, proprio in quanto indirizzati alla gestione di problematiche di integrazione “general purpose”, si tratta di ambienti significativamente complessi, tanto da manifestare alcune controindicazioni per l'implementazione dell'ambiente SPCoop:

1. La configurazione di nuovi servizi è in generale piuttosto complessa e richiede solitamente la scrittura di codice ad-hoc per programmare le logiche di trasformazione. Questa complessità aumenta il rischio di errori nella configurazione dei nuovi servizi.
2. Le funzionalità di trasformazione di messaggio dei Service Bus non sono sufficienti a gestire il protocollo SPCoop, che prevede complesse logiche di indirizzamento e affidabilità. Quindi, anche usando un ESB rimane necessario implementare il servizio SPCoop come servizio di base, parte del core del Service Bus.
3. L'uso di una piattaforma “general purpose” di questo tipo rende difficile assicurare performance elevate in termini di numero di messaggi gestibili al secondo. Questo problema è particolarmente serio per quelle istanze di Porte di Dominio che operano come smistatori e validatori di messaggi in transito, come ad esempio si può verificare per dei gateway interregionali. Queste porte hanno solo funzionalità di cooperazione e non di integrazione, e sono proprio quelle soggette al maggior carico di messaggi da smistare.
4. Gli ESB supportano sempre connettori SOAP su vari trasporti standard (come SMTP, http e https); tuttavia le funzionalità a maggior valore aggiunto sono specifiche per la piattaforma sulla quale sono implementati (ad esempio disponibilità del trasporto JMS per gli ESB su piattaforma J2EE). Quindi la dipendenza dalle funzionalità a valore aggiunto degli ESB implica spesso anche la dipendenza da una specifica architettura software.

### ***Approccio “Web Services Mediator”***

Un nuovo approccio per la soluzione della stessa problematica affrontata dagli ESB è emerso di recente in alcuni progetti, tra i quali sta raggiungendo una notevole notorietà il progetto Apache Synapse (<http://ws.apache.org/synapse/>). In questo approccio si parte dalla constatazione che l'ampia diffusione dell'XML e del paradigma dei Web Services permette di considerare i sistemi interni al Dominio di Servizio come già capaci o facilmente adattabili al dialogo tramite Web Services. Se si assume questa premessa, la componente di integrazione del Service Bus non dovrà più supportare tecnologie diverse per ogni possibile sistema legacy da interfacciare (CORBA, RMI, JMS, .NET, etc.), ma potrà essere invece un generico container che funga da mediatore dei messaggi SOAP in arrivo dai clienti interni per i servizi esterni e viceversa.

---

<sup>2</sup> <http://ws.apache.org/axis/>, <http://ws.apache.org/axis2/>, <http://xfire.codehaus.org/>.

<sup>3</sup> <http://mule.codehaus.org/>, <http://incubator.apache.org/servicemix/>, <http://www.ionaceltix.com/>, <http://www.iona.com/products/artix>.

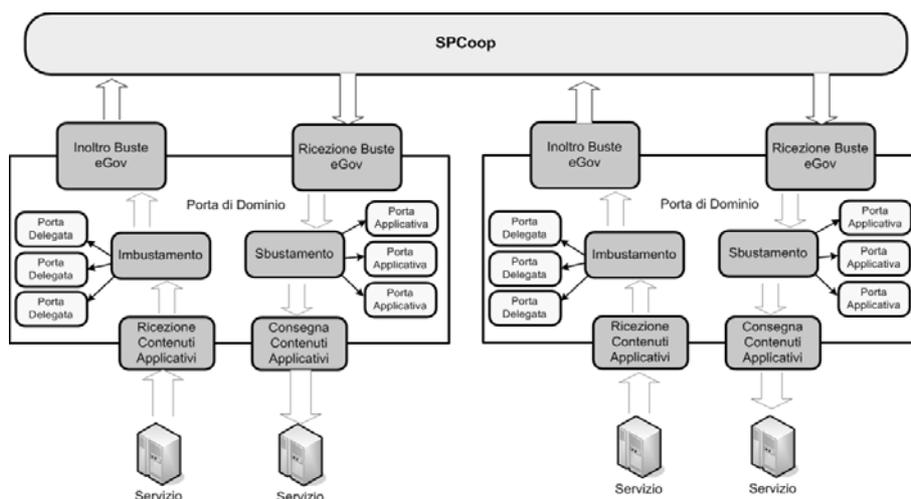
Il mediatore potrà ancora fornire tutti i servizi a valore aggiunto dei Service Bus, introducendo o interpretando gli header del messaggio SOAP in transito, ma rispetto agli ESB tradizionali esporrà solo un generico connettore per l'accettazione di buste SOAP, come componente di integrazione.

Dal punto di vista dell'architettura Web Services, il ruolo del Web Services Mediator è quello di un "nodo intermedio". L'architettura dei Web Services prevede infatti che un messaggio SOAP viaggi dal mittente fino al destinatario finale, passando eventualmente attraverso un certo numero di nodi intermedi. Ogni nodo intermedio quindi riceverà il messaggio SOAP e lo ritrasmetterà verso la destinazione finale. Mentre il contenuto applicativo (il body) è di competenza esclusivamente del destinatario finale, le informazioni contenute negli header del messaggio possono essere destinate e/o generate anche dai nodi intermedi.<sup>4</sup> Di conseguenza quello del Web Services Mediator si pone anche come l'approccio più corretto da un punto di vista architetturale.

#### 4. La Porta di Dominio in OpenSPCoop

Le varie soluzioni di Porta di Dominio SPCoop rese disponibili nel tempo sono tutte realizzate come framework che forniscono soluzioni del tipo *end to end* o soluzioni di tipo ESB. Abbiamo già elencato le controindicazioni di questi approcci che risultano particolarmente evidenti nell'architettura SPCoop, dove la necessità di implementare i componenti di integrazione sulla stessa Porta di Dominio entra in conflitto con le criticità, soprattutto di sicurezza, della porta stessa.

La Porta di Dominio di OpenSPCoop, in discontinuità con gli altri prodotti disponibili, ha anticipato l'adozione di un approccio di tipo *Web Services Mediator*, la cui validità è stata successivamente dimostrata anche dal progetto Synapse, riuscendo così ad aumentare gli standard di sicurezza e affidabilità della PdD e a ridurre drasticamente l'impatto e quindi il costo dell'adeguamento ad SPCoop da parte degli Enti Pubblici. Grazie a quest'approccio, infatti, l'abilitazione di nuove porte delegate e/o applicative non richiede la programmazione e il deploy di componenti applicativi ad-hoc, come normalmente richiesto da altre implementazioni della Porta di Dominio, ma unicamente la descrizione, in un apposito formato xml, delle modalità di trattamento dei messaggi in transito da parte della Porta di Dominio, e l'inserimento di tali descrizioni in un opportuno repository.



<sup>4</sup> Nonostante il concetto di nodo intermedio sia presente sin dalla prima versione della specifica SOAP, i Web Services sono stati sinora utilizzati quasi unicamente come tecnologia di comunicazione end-to-end, in cui mittente e destinatario interagiscono direttamente. Anche i servizi dell'architettura Web Services, come ad esempio WS-Security e WS-Reliability, sono tipicamente gestiti direttamente dai nodi mittente e destinatario finale.

La soluzione, la cui architettura è illustrata nella figura precedente, ha il grande vantaggio di non richiedere più la realizzazione di uno specifico componente di integrazione per ogni servizio da utilizzare (porta delegata) o da fornire (porta applicativa), riducendo così drasticamente la complessità di attivazione di nuovi servizi SPCoop.

La PdD di OpenSPCoop è quindi in grado di accettare generiche buste SOAP dall'interno del dominio, occupandosi in maniera trasparente rispetto agli applicativi della gestione degli header previsti dalla specifica SPCoop. Eventuali aspetti di integrazione più specifici (ad esempio l'imbustamento in SOAP di contenuti applicativi legacy), laddove necessari, restano a carico di ulteriori componenti di integrazione interni al dominio, come possono essere ESB di tipo tradizionale o applicazioni wrapper ad-hoc.

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

Differentemente da quanto avviene per il componente di cooperazione della Porta di Dominio (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 attualmente 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 PdD 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 che il servizio applicativo utilizzi 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.

## **5. Il Registro Servizi di OpenSPCoop**

L'interpretazione dinamica delle buste in transito da parte della Porta di Dominio è possibile grazie alla presenza del Registro dei Servizi di OpenSPCoop. Il Registro permette la registrazione e la successiva interrogazione degli Accordi di Servizio (AS) della specifica SPCoop, i quali contengono i riferimenti ai soggetti erogatori e fruitori dei Servizi, alle interfacce dei servizi erogati e fruiti, alle politiche di sicurezza, ai *service level agreements*, e 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 definisce 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>.

## **6. Il Gestore Eventi di OpenSPCoop**

OpenSPCoop comprende un'applicazione Web Service che realizza lo scenario di coordinamento ad eventi secondo l'architettura EDA. Si tratta di un'applicazione web services, integrata con la Porta di Dominio tramite l'uso di apposite porte delegate e applicative.

Mentre la pubblicazione di un messaggio risulta 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 iscritti alla notifica dell'evento richiede la spedizione di una diversa copia dello stesso messaggio ad ogni iscritto. 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'identificatore eGov del messaggio da spedire e non tutto 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.

## **7. Open Source nel Software Custom per le P.A.: una Opzione Possibile?**

Come accennato nell'Introduzione, sono ben noti i vantaggi di cui godono i progetti open source come OpenSPCoop, in termini di interoperabilità, sicurezza e innovazione del codice, e di potenzialità di aggregazione di una comunità di sviluppatori e utenti. Questi aspetti qualitativi, prima ancora degli eventuali vantaggi economici derivanti dal suo utilizzo, dovrebbero rendere il software open source particolarmente appetibile per le Pubbliche Amministrazioni, poiché consentono di soddisfare in modo immediato alcuni requisiti specifici delle P.A..

Ricordiamo tra questi la necessità di archiviazione di dati (ad esempio anagrafici o censuari) a lungo termine, che è garantita dall'uso di formati aperti e standard, le esigenze di privacy e di protezione di dati sensibili, garantite almeno in parte dalla verificabilità del codice sorgente, e più generalmente l'obbligo istituzionale di trasparenza nei confronti dei cittadini. Inoltre la natura distribuita e solo parzialmente gerarchica, ma principalmente autonoma delle P.A., sembra sposarsi bene con un modello di sviluppo software basato su di una comunità di attori, in cui esse possono proporsi come utenti, come consulenti per loro specifiche problematiche, o anche come sviluppatori. Infine è evidente come una soluzione open source adottata da una P.A. per proprie

esigenze si presti ad essere riusata liberamente, tale e quale oppure con opportune modifiche, da altre P.A. aventi necessità analoghe.

Queste semplici considerazioni spiegano le iniziative intraprese da alcuni governi, europei e non, miranti a favorire la diffusione della cultura dell'open source nelle P.A., nonché, in un paese come la Francia dove questa cultura è già ben radicata, il successo di un'associazione di P.A. come l'ADULLACT (<http://www.adullact.org>), avente l'obiettivo di sviluppare e mantenere "un patrimonio comune di software libero utile alla missione del servizio pubblico". Per quanto riguarda l'Italia, invece, purtroppo a tutt'oggi le iniziative governative nei confronti dell'open source non sono univoche, nonostante considerazioni come quelle riportate sopra siano analizzate in dettaglio in importanti documenti come [13], e nonostante le numerose dichiarazioni di intenti di non voler penalizzare pregiudizialmente soluzioni open source nelle pubbliche commesse. Al contrario, riteniamo che nei bandi di gara per lo sviluppo di software custom pubblicati negli ultimi anni dalle P.A., e anche nella formulazione dei progetti di riuso che proprio in questi mesi si stanno finalizzando, siano previste delle condizioni che rendono a priori non proponibili alcune soluzioni basate su software open source. Più precisamente, la maggior parte dei bandi richiede che il fornitore del software custom ceda alla P.A. committente la proprietà del software sviluppato, e analogamente i progetti di riuso richiedono che il software che una P.A. cedente offre ad altre P.A. per il riuso sia di sua proprietà. Prima di analizzare l'effetto negativo di queste condizioni sull'open source e di proporre delle soluzioni alternative, riteniamo istruttivo analizzarne la genesi.

Come risaputo, la "Commissione per il software a codice sorgente aperto nella Pubblica Amministrazione" istituita dal Ministro per l'Innovazione e le Tecnologie Lucio Stanca alla fine del 2002, ha pubblicato nel maggio 2003 una "Indagine conoscitiva" [13] che ha sancito alcuni principi ripresi successivamente dalla cosiddetta "Direttiva Stanca" emanata dal Ministro alla fine del 2003 [14]. Nell'"Indagine conoscitiva", la Commissione identifica due "macrotipologie di prodotti" software rilevanti per la P.A.: i pacchetti, come le suite per office automation e i DBMS, e il software custom, sviluppato ad hoc da un fornitore per una specifica esigenza di una o più P.A.. Anche se tale classificazione non è presentata come esaustiva, nel resto del documento vengono considerate solo queste due tipologie.

In particolare, per il software custom, che è l'unico rilevante per la presente discussione, la Commissione suggerisce che questo software (e le personalizzazioni) debba essere di piena proprietà (non necessariamente esclusiva) della P.A. committente, e questo deve essere garantito da opportune clausole contrattuali. Questa piena proprietà garantisce alla P.A. caratteristiche di primaria importanza come la modificabilità, la verificabilità e la possibilità di riuso di tale software.

Queste indicazioni sono state completamente recepite nella Direttiva Stanca, che nell'Art. 5 recita: *Nel caso di programmi informatici sviluppati ad hoc, l'amministrazione committente acquisisce la proprietà del prodotto finito, avendo contribuito con proprie risorse all'identificazione dei requisiti, all'analisi funzionale, al controllo e al collaudo del software realizzato dall'impresa contraente. Sarà cura dei committenti inserire, nei relativi contratti, clausole idonee ad attestare la proprietà dei programmi.*

Il principio che il software custom debba divenire di proprietà della P.A. committente viene ribadito con chiarezza nelle attuali iniziative del CNIPA nell'ambito delle attività per il supporto alla diffusione dell'e-government per le Regioni e gli Enti locali. Infatti, tra le cinque linee di azione della seconda fase di tale attività [15], quella di maggiore rilievo in termini di budget allocato riguarda i progetti di riuso di soluzioni di e-government, che ne prevedono il passaggio da una P.A. cedente ad una o più P.A. riceventi. L'"Avviso per la realizzazione di progetti di riuso" pubblicato nel febbraio 2007 [16] prevede risorse per 60 milioni di Euro per i progetti di riuso, e richiede che le soluzioni di e-Government proposte per il riuso debbano "essere di proprietà e nella piena fruibilità della pubblica amministrazione cedente". Ancora più esplicitamente, nella versione acceduta il

5/5/07, l'Art. 4 dello "Schema tipo di un contratto di riuso" pubblicato sulle pagine del CNIPA recita, al primo comma: *L'amministrazione concedente garantisce che il programma è di propria esclusiva proprietà e che il perfezionamento del presente atto non costituisce violazione di diritti di titolarità di terzi*, suggerendo, forse in modo indebito, che persino una *proprietà non esclusiva* possa non essere sufficiente per accedere ai finanziamenti per i progetti di riuso.

Come anticipato, in accordo con le indicazioni dei documenti citati, gli attuali bandi per la fornitura di software custom e i contratti per i progetti di riuso contengono clausole che garantiscono che la P.A. committente/cedente sia proprietaria del software in questione. Ma questi requisiti sono incompatibili con gran parte delle licenze open source, che non permettono il trasferimento della proprietà del software.

Questo problema non è puramente teorico. In uno scenario del tutto realistico, una P.A. potrebbe commissionare lo sviluppo di una soluzione di e-government ad un fornitore. Quest'ultimo, analizzando le varie soluzioni possibili, potrebbe individuarne una che consiste nella personalizzazione di un sistema disponibile con licenza GPL (o simile). In questo caso il fornitore avrebbe titolo di trasferire al committente solo la proprietà delle personalizzazioni effettuate, visto che la proprietà del sistema originale appartiene ai vari sviluppatori della comunità, che detengono il copyright dei singoli componenti realizzati. Pertanto, anche se quella soluzione fosse risultata la più conveniente ad un'analisi dei costi, il fornitore non avrebbe potuto proporla. E' evidente inoltre come questo meccanismo inibisca le possibilità di riuso del software esistente (il sistema originale, in questo caso): sembra quindi che richiedere la proprietà del software commissionato, anche se da un lato garantisce alla P.A. committente la possibilità di cederlo ad altre P.A. in riuso, dall'altro ostacola altre forme di riuso in uno scenario più ampio. Si osservi infine che se una P.A. decidesse di partecipare ad un progetto open source per lo sviluppo di una soluzione di e-government per soddisfare una propria esigenza, non potrebbe successivamente accedere ai finanziamenti per un progetto di riuso per tale soluzione, mancando il requisito di proprietà.

E' nostra convinzione che la soluzione dei problemi individuati consista nel rimuovere i vincoli di proprietà già ampiamente discussi. I bandi e i progetti di riuso non dovrebbero quindi imporre vincoli pregiudiziali di proprietà sul software in questione, ma dovrebbero solo fissare i vincoli funzionali e strategici, cioè i "diritti minimi" che la P.A. interessata deve poter esercitare su tale software per garantirne la totale fruibilità e la possibilità di concederlo in riuso. Solo in questo modo le soluzioni di e-government basate sull'open source potrebbero godere di pari opportunità rispetto a soluzioni proprietarie.

Riteniamo corretto sottolineare a questo punto che forse il problema individuato non è passato inosservato al Gruppo di Lavoro su "Codice sorgente aperto" istituito nel febbraio 2004, che ha proseguito i lavori della precedente Commissione. Infatti nel Rapporto Conclusivo [17] vengono analizzate e approfondite sia le conclusioni della Commissione che le disposizioni della Direttiva Stanca, ma non viene mai citato l'Art. 5 riportato sopra. In particolare, a fronte di circa venti pagine che propongono precisi criteri di valutazione delle offerte di pacchetti proprietari o open source, per la valutazione del software custom vengono solo enunciati in una pagina alcuni principi generali, senza peraltro ribadire il vincolo di proprietà della P.A. committente, e rimandando ulteriori approfondimenti al costituendo Centro di Competenza.

Per completare il quadro, aldilà dei vari documenti programmatici è d'uopo commentare la normativa formalmente vigente. Nella versione attuale del Codice dell'Amministrazione Digitale [18], il riuso di software nelle P.A. è regolamentato dall'Art. 69, *Riuso dei programmi informatici*, il cui terzo comma così recita: *Le pubbliche amministrazioni inseriscono, nei contratti per l'acquisizione di programmi informatici, di cui al comma 1, clausole che garantiscano il diritto di disporre dei programmi ai fini del riuso da parte della medesima o di altre amministrazioni.*

Come evidente dalla lettura dell'articolo e di questo comma in particolare, la normativa non prescrive esplicitamente che la P.A. committente richieda per contratto la proprietà del software commissionato, ma solo i diritti necessari per consentirne il riuso. E' auspicabile che in tempi brevi gli organismi preposti possano intervenire con chiarificazioni in merito. Allo stato attuale, infatti, si prospetta uno scenario in cui legittimi ricorsi possano essere sollevati contro bandi di gara che penalizzino aprioristicamente soluzioni open source, imponendo vincoli non prescritti per legge.

## 8. Bibliografia

- [1] Andrea Corradini, Tito Flagella e Andrea Poli, *Aspetti di Interoperabilità della Specifica SPCoop nell'implementazione OpenSPCoop 1.0*, PAAL 2007.
- [2] SPC, "Sistema pubblico di cooperazione: Architettura, V. 1.0", CNIPA, 25/11/2004.
- [3] SPC, "Sistema pubblico di cooperazione: Porta di Dominio, V. 1.0", CNIPA, 14/10/2005.
- [4] SPC, "Specifiche della Busta di e-Government, Edizione 1.0", CNIPA, 21/04/2004.
- [5] SPC, "Sistema pubblico di cooperazione: Busta di e-Gov, V. 1.1", CNIPA, 14/10/2005
- [6] SPC, "Sistema pubblico di cooperazione: Accordo di Servizio, V.1.0", CNIPA, 14/10/2005.
- [7] SPC, "Sistema pubblico di cooperazione: Servizi di Registro, V. 1.0", CNIPA, 14/10/2005.
- [8] Ruggero Barsacchi, *Progettazione di un framework Open Source per la cooperazione applicativa nella Pubblica Amministrazione*, Tesi di Laurea Specialistica in Tecnologie Informatiche, Università di Pisa, Luglio 2005.
- [9] Andrea Poli, *OpenSPCoop: un'implementazione della Specifica di Cooperazione Applicativa per la Pubblica Amministrazione Italiana*, Tesi di Laurea Specialistica in Tecnologie Informatiche, Università di Pisa, Febbraio 2006.
- [10] 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.
- [11] Web Services Interoperability Organization, *Basic Profile Version 1.1*, <http://www.wsi.org/Profiles/BasicProfile-1.1-2004-08-24.html>.
- [12] S. Vinoski, *Integration with Web Services*, IEEE Internet Computing 7(6), 2003, pp. 75-77, <http://csdl.computer.org/comp/mags/ic/2003/06/w6075abs.htm>.
- [13] Commissione per il software a codice sorgente aperto nella Pubblica Amministrazione, *Indagine conoscitiva su software a codice aperto nella Pubblica Amministrazione*, maggio 2003, [http://www.cnipa.gov.it/site/\\_files/indagine\\_commissione\\_os.pdf](http://www.cnipa.gov.it/site/_files/indagine_commissione_os.pdf).
- [14] Ministro per l'Innovazione e le Tecnologie, *Sviluppo ed utilizzazione dei programmi informatici da parte delle pubbliche amministrazioni*, Direttiva del 19 dicembre 2003, G.U. n. 31 del 7-2-2004, <http://www.innovazione.gov.it/ita/normativa/allegati/Dir191203.pdf>.
- [15] Ministro per l'Innovazione e le Tecnologie, *L'e-government nelle Regioni e negli Enti locali: II fase di attuazione*, Documento approvato dalla Conferenza Unificata Stato, Regioni, Città e Autonomie Locali il 27 novembre 2003, [http://www.cnipa.gov.it/site/\\_files/egov\\_Fase2.pdf](http://www.cnipa.gov.it/site/_files/egov_Fase2.pdf).
- [16] Presidente del CNIPA, *Avviso per la realizzazione di progetti di riuso*, G.U. n. 30 del 6-2-2007, [http://www.cnipa.gov.it/site/\\_files/Avviso%20RIUSO%20012007.pdf](http://www.cnipa.gov.it/site/_files/Avviso%20RIUSO%20012007.pdf).
- [17] Gruppo di lavoro "Codice sorgente aperto" ("Open Source") del CNIPA, *Rapporto conclusivo*, maggio 2004, [http://www.cnipa.gov.it/site/\\_files/Rapporto%20conclusivo\\_OSS.pdf](http://www.cnipa.gov.it/site/_files/Rapporto%20conclusivo_OSS.pdf)
- [18] Codice dell'Amministrazione Digitale, D.L. 7 marzo 2005, n. 82, modificato col D.L. 4 aprile 2006, n. 159, G.U. n. 99 del 29-4-2006.