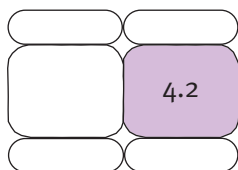




OPEN SOURCE E PUBBLICA AMMINISTRAZIONE

Giovanna Sissa



Imprese, mondo della ricerca, utenti di tecnologie informatiche e telecomunicazioni, Pubbliche Amministrazioni guardano con interesse al software open source e al modello di sviluppo a esso collegato. Da molte parti si ritiene che il paradigma open source possa favorire il rilancio dell'industria informatica nazionale ed europea e il dibattito sul software open source concerne l'impatto che esso può avere sullo sviluppo del Sistema Paese.

1. LA STORIA

L'origine di quel complesso fenomeno che va sotto il nome di *Open Source Software* (OSS) è intimamente connessa con la storia del sistema operativo UNIX.

La comunità scientifica creatasi con la prima fase di sviluppo di UNIX quando tramontò l'idea di un suo sviluppo non proprietario aveva maturato al suo interno una forte coesione intellettuale e un modello culturale di collaborazione. Si ricostituì, infatti, parzialmente nella **Free Software Foundation** (FSF) [4] e poi nella **Open Source Initiative** (OSI) [12]. I due movimenti che ne sono derivati, quello del *software libero* e quello del codice a *sorgente aperto*, condividono entrambe l'idea guida della *disponibilità completa e senza limitazioni dei sorgenti*, ma si diversificano, oltre che per la terminologia, per alcuni aspetti filosofici e di principio [8, 9].

I sostenitori del software libero scelgono questo termine per insistere sulle "libertà" associate a un software: tali *libertà* sono dei *criteri*, ma soprattutto delle *motivazioni etiche*. I sostenitori dell'uso del termine "codice

a sorgente aperto" insistono sulle caratteristiche operative, e in particolare sulla disponibilità del codice sorgente, senza tentare di fornire motivazioni etiche.

Fatte salve queste precisazioni, ricche peraltro di implicazioni [4, 8, 12], per non appesantire il lettore meno esperto e per concentrarsi, invece, sulle specificità nella Pubblica Amministrazione, nel seguito di questo articolo si useranno indifferentemente i termini "open source software" e "software libero" per indicare l'insieme delle tematiche e delle proposte che si rifanno alle interpretazioni citate.

2. IL CUORE DELL'OPEN SOURCE: LE LICENZE

In funzione del livello di contribuzione richiesto all'utente (gratuità), di disponibilità del codice sorgente (trasparenza) o di importanza data alla nozione di proprietà (diritti d'autore), i software sono designati con i termini di *dominio pubblico*, *liberi*, *freeware*, *shareware* o *proprietary*. È la natura della licenza

Per la **Free Software Foundation** (FSF), un software è considerato libero se la licenza accorda all'utilizzatore le seguenti quattro libertà:

- *Libertà di eseguire il programma per qualunque uso.*
- *Libertà di studiare il funzionamento del programma e di adattarlo ai propri bisogni. Per questo l'accesso al codice sorgente è condizione necessaria.*
- *Libertà di ridistribuire delle copie.*
- *Libertà di migliorare il programma e di pubblicare le modifiche, per farne profittare tutta la comunità di utenti e di sviluppatori.* A tal fine è condizione indispensabile l'accesso al codice sorgente.

La FSF ha creato il concetto di "copyleft". Questo termine definisce una licenza che riprende le quattro libertà suddette e i cui termini devono essere ripresi in modo identico in caso di nuova distribuzione. Ciò permette di evitare che una distribuzione di software modificato restringa i diritti iniziali.

I criteri, invece, dell'**Open Source Initiative** (OSI) che permettono di determinare la natura libera o proprietaria di una licenza di software si articolano in nove punti:

- I** *Libera ridistribuzione*
- II** *Codice sorgente*
- III** *Prodotti derivati*
- IV** *Integrità del codice sorgente dell'autore*
- V** *Assenza di discriminazione nei confronti di persone o gruppi*
- V** *Assenza di discriminazione nei confronti di sfere di attività*
- VII** *Distribuzione di licenza*
- VIII** *La licenza non deve essere specifica di un prodotto*
- IX** *La licenza non deve imporre limitazioni ad altri software: ovvero, esigere che gli altri programmi distribuiti sullo stesso supporto fisico siano anch'essi software liberi.*

associata al software a determinare l'appartenenza a una delle categorie suddette. Il concetto di software libero è, quindi, associato all'insieme di software coperti da un tipo di licenza particolare: le licenze di software libero [4, 12].

Giuridicamente, il software libero non è un software senza diritti: esso resta governato dalle disposizioni della licenza e l'autore del software resta il titolare dell'insieme dei diritti di autore.

Le licenze di software libero consistono in una messa a disposizione del software con l'intento di permettere la libera evoluzione del software medesimo. La licenza non ha per scopo il trasferimento di un diritto di proprietà o la rinuncia al diritto di autore o di "far cadere" il software nel pubblico dominio; diffondendo il proprio software libero l'autore può assicurarsi che la libera utilizzazione del software non sia perturbata dalle azioni dei soggetti alla licenza.

Richard Stallman e la FSF [9] definiscono il concetto di *copyleft* in contrapposizione al tradizionale *copyright*: ove il copyright tende a tutelare il diritto d'autore, anche attraverso limitazioni all'accesso della conoscenza, mentre il copyleft intende tutelare il più generale diritto della collettività a fruire dei prodotti dell'innovazione.

I principi del copyleft vengono formalizzati

dalla FSF nella cosiddetta *General Public License* (GPL). Il cliente di accordo GPL è vincolato a utilizzare a sua volta la GPL e dovrà, quindi, fornire il codice sorgente delle estensioni realizzate. In altre parole, la GPL è un modello di licenza ricorsivo. Il codice e le "libertà" a esso associate diventano così legalmente inseparabili.

Il mondo dell'open source non coincide con la GPL. Esistono modelli di licenza alternativi quali ad esempio, LGPL (*Lesser General Public License*), Artistic, BSD (*Berkeley Software Distribution*), MPL (*Mozilla Public License*) ecc., che prevedono, in forme differenti, l'apertura del codice sorgente.

Alcune licenze, come la GPL, vietano la realizzazione di soluzioni proprietarie a partire dal software libero. Altre licenze, come le licenze tipo MIT o BSD, consentono le derivazioni proprietarie. Questa possibilità offerta allo sfruttamento proprietario di una base di codice costituisce la principale distinzione fra le differenti licenze di software libero.

Secondo l'Open Source Initiative trentuno licenze rispondono ai criteri dell'associazione stessa (Tabella 1).

Le licenze OSS non prescrivono che il software debba essere ceduto gratuitamente. Il software open source non è alternativo al software commerciale: il modello OSS non preclude la presenza di distribuzione com-

Apache Software License	http://www.opensource.org/licenses/apachepl.html
Apple Public Source License	http://www.opensource.org/licenses/apsl.html
Artistic license	http://www.opensource.org/licenses/artistic-license.html
BSD license	http://www.opensource.org/licenses/bsd-license.html
Common Public License	http://www.opensource.org/licenses/cpl.html
Eiffel Forums License	http://www.opensource.org/licenses/eiffel.html
GNU General Public License (GPL)	http://www.opensource.org/licenses/gpl-license.html
GNU Library or "Lesser" Public License (LGPL)	http://www.opensource.org/licenses/lgpl-license.html
IBM Public License	http://www.opensource.org/licenses/ibmpl.html
Intel Open Source License	http://www.opensource.org/licenses/intel-open-source-license.html
Jabber Open Source License	http://www.opensource.org/licenses/jabberpl.html
MIT license	http://www.opensource.org/licenses/mit-license.html
MITRE Collaborative Virtual Workspace License (CVW License)	http://www.opensource.org/licenses/mitrepl.html
Motosoto License	http://www.opensource.org/licenses/motosoto.html
Mozilla Public License 1.1 (MPL 1.1)	http://www.opensource.org/licenses/mozilla1.1.html
Nethack License	http://www.opensource.org/licenses/nethack.html
Nokia Open Source License (NOKOS License) Version 1.0a	http://www.opensource.org/licenses/nokia.html
Open Group Test Suite License	http://www.opensource.org/licenses/opengroup.html
Python license (CNRI Python License)	http://www.opensource.org/licenses/pythonpl.html
Python Software Foundation License	http://www.opensource.org/licenses/PythonSoftFoundation.html
Qt Public License (QPL)	http://www.opensource.org/licenses/qtpl.html
Ricoh Source Code Public License	http://www.opensource.org/licenses/ricohpl.html
Sleepycat License	http://www.opensource.org/licenses/ricohpl.html
Sun Industry Standards Source License (SISSL)	http://www.opensource.org/licenses/sisslpl.html
Sun Public License	http://www.opensource.org/licenses/sunpublic.html
University of Illinois/NCSA Open Source License	http://www.opensource.org/licenses/UoI-NCSA.html
Vovida Software License v. 1.0	http://www.opensource.org/licenses/vovidapl.html
W3C License	http://www.opensource.org/licenses/W3C.html
X.Net License	http://www.opensource.org/licenses/xnet.html
zlib/libpng license	http://www.opensource.org/licenses/zlib-license.html
Zope Public License	http://www.opensource.org/licenses/zpl.html

TABELLA 1

Le trentuno licenze che rispondono ai criteri dell'Open Source Initiative



merciale, di fornitori di valore aggiunto o di servizi di supporto.

È corretto definire l'OSS come alternativa al modello di licenza proprietario (*closed source*), in cui l'accesso al codice sorgente non è concesso e il fornitore del software vende all'utente una "licenza d'utilizzo", temporanea o illimitata, che consente l'uso del prodotto, ma non implica in nessun modo che l'utente acquisisca la proprietà del software.

3. I MATTATORI DELL'OPEN SOURCE

Il grosso pubblico ha preso familiarità con il concetto di software libero dopo il successo del sistema Linux, un sistema operativo *Unix-like* il cui *kernel* fu realizzato nel 1991 da Linus Torvalds e da un folto gruppo di collaboratori volontari sparsi in tutto il mondo. Questo kernel, in virtù di un set di componenti "liberi" rilasciati sotto la licenza GPL del progetto GNU (*Gnu's Not Unix*), costituisce ora un software completamente libero. Per estensione e semplificazione il sistema viene chiamato LINUX (più esattamente è il sistema GNU/Linux).

GNU/Linux viene distribuito da alcuni facilitatori commerciali, che forniscono servizi di supporto e *package* a basso costo, definiti comunemente "distribuzioni".

Gli esempi più famosi e citati di OSS includono, oltre a Linux, il *server web* Apache [10] che è l'OSS più diffuso, anche in quanto disponibile su vari sistemi operativi: l'ultima rilevazione di *Netcraft Web Server Survey* [11] di febbraio 2003 segnala che Apache ha raggiunto il 66,75% di presenza su circa 35 milioni di siti analizzati.

Da citare, inoltre, BIND, che opera la risoluzione del DNS per l'intera Internet, Sendmail, il più importante e diffuso software di trasporto e-mail su Internet e Openoffice.org, la suite open source di *office automation*.

3.1. I motori della produzione del software open source

Le entità che contribuiscono a livello mondiale alla produzione del software libero afferiscono sia al settore pubblico che privato.

Nel settore pubblico, i primi contributori so-

no stati i laboratori di ricerca e le università, che sono stati poi affiancati dalle amministrazioni centrali e locali. Per un ricercatore di un laboratorio o università, il quadro del software libero permette di diffondere dei lavori innovativi a una ampia comunità o anche di fornire un contributo personale che va ad aggiungersi a un insieme già elaborato. Il contributo del ricercatore resta identificabile e può accrescerne la notorietà. Per le amministrazioni centrali e locali, una delle motivazioni è la messa in opera di interfacce o di strumenti utilizzabili da entità connesse, ma non in rapporto gerarchico.

Un fenomeno diffuso nell'OSS è la cosiddetta *Advocacy*, teorizzata dalle figure di spicco del movimento. Si tratta di una sorta di *marketing ono-to-one*, in base al quale gli utenti dei programmi OSS sono invitati a convincere altri membri del loro collettivo di riferimento a fare altrettanto e ad abbandonare il mondo commerciale. Il software open source costituisce, quindi, un paradigma di sviluppo, di diffusione e di cooperazione nel campo della *Information Technology* (IT).

4. STATO DELL'ARTE DELL'OPEN SOURCE SOFTWARE NELLA PUBBLICA AMMINISTRAZIONE

I dati relativi all'uso di OSS nella Pubblica Amministrazione (PA) sono pochi e frammentari. Come già detto, le informazioni più dettagliate riguardano l'uso di Apache come HTTP (*Hyper Text Transfer Protocol*) server.

I report sull'argomento di alcuni analisti, quali Forrester, META, IDC o Gartner, sono contraddittori a causa della complessità del rilevamento delle informazioni e in quanto non esistono dati per il solo settore pubblico.

La Commissione Europea ha promosso alcuni studi sul tema. I più articolati e significativi sono senz'altro *Study into the use of Open Source Software in the Public Sector - An IDA Study - Interchange of Data between Administrations* [2] e *FLOSS (Free/Libre and Open Source Software: Survey and Study)* [7], ai quali molta parte del presente articolo si ispira.

Le amministrazioni pubbliche nei vari stati, e

in Europa, in particolare (Figura 1), hanno propensioni e *policy* sul tema dell'OSS molto diverse: dalla forte adesione di Francia e Germania, al pragmatismo inglese fino allo scarso interesse di altri. L'attenzione sembra, comunque, in ascesa ovunque.

4.1. L'open source in Europa

La citata ricerca dalla Commissione Europea IDA Project [2] riporta i dati di una rilevazione effettuata mediante un questionario compilato dai responsabili dei sistemi informativi delle pubbliche amministrazioni dei paesi comunitari. Anche se la rilevazione presenta limiti quantitativi, come evidenziato nel report stesso poiché si tratta degli unici dati a livello europeo relativi al settore pubblico, tali risultati sono comunque interessanti.

Risulta che l'utilizzo di OSS nella PA è concentrato nella fascia "server", dove Linux spesso completa o sostituisce precedenti versioni proprietarie di Unix. Risulta vincente il duo "Apache/Linux".

Sul lato *client*, invece, le suite di Office Automation OSS costituiscono il fenomeno più interessante, anche se con dimensioni quanti-

tativamente ben più contenute rispetto a Linux nella fascia server.

4.2. Alcuni dati della Pubblica Amministrazione Locale italiana

Alcuni ricercatori dell'Osservatorio P.A.O.S. (*Software Opensource nella Pubblica Amministrazione* -Università di Bologna) effettuano una rilevazione sistematica relativa al tipo di server HTTP utilizzato dalle PAL italiane [3] (Figura 2). Tale rilevazione è effettuata "interrogando", per mezzo di una procedura automatizzata, i server relativi a una lista di domini riservati alla amministrazione pubblica locale e riguarda esclusivamente i siti dei nomi di dominio di comuni, province e regioni.

Da un confronto con i risultati della citata rilevazione periodica realizzata da Netcraft sull'intera popolazione dei siti Web mondiale emerge un utilizzo relativamente ridotto dei server (open source) Apache nell'amministrazione pubblica italiana: meno del 40% nella PA italiana contro oltre il 60% nel mondo.

La rilevazione indica una stima della percentuale di adozione dei diversi tipi di server HTTP all'interno di comuni, province e regioni

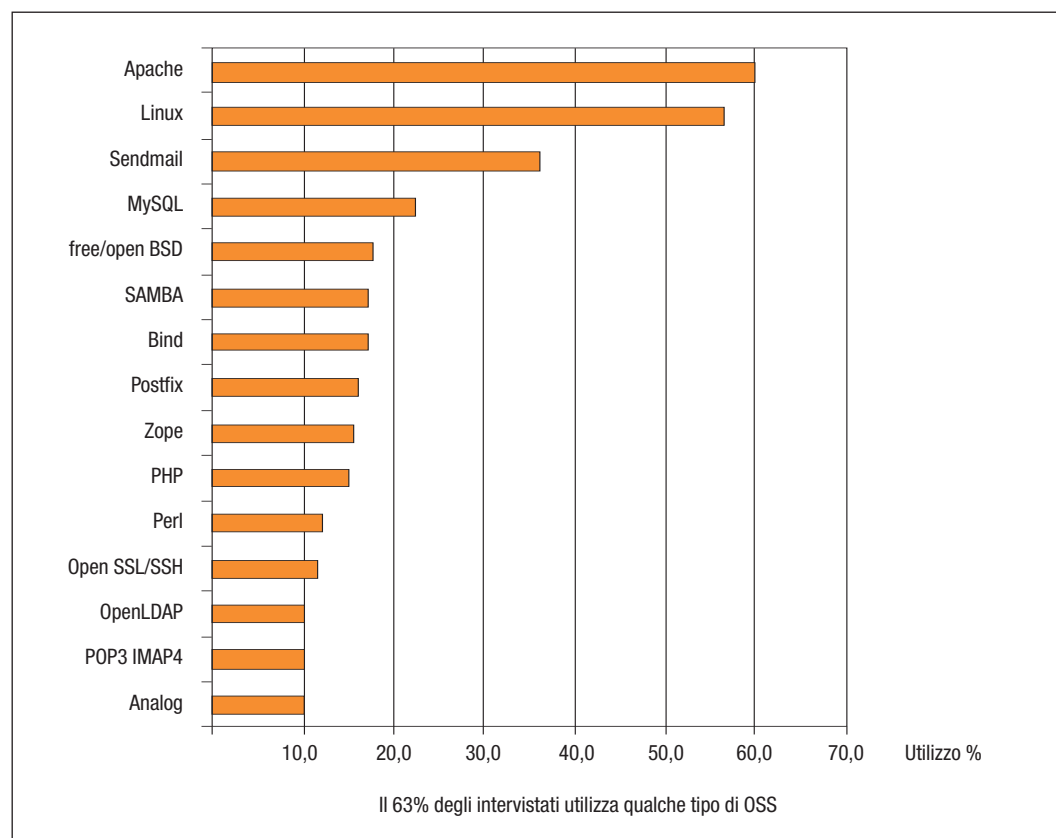


FIGURA 1
OSS nelle Pubbliche Amministrazioni europee – lato server (Fonte IDA study, European Commission, 2001)

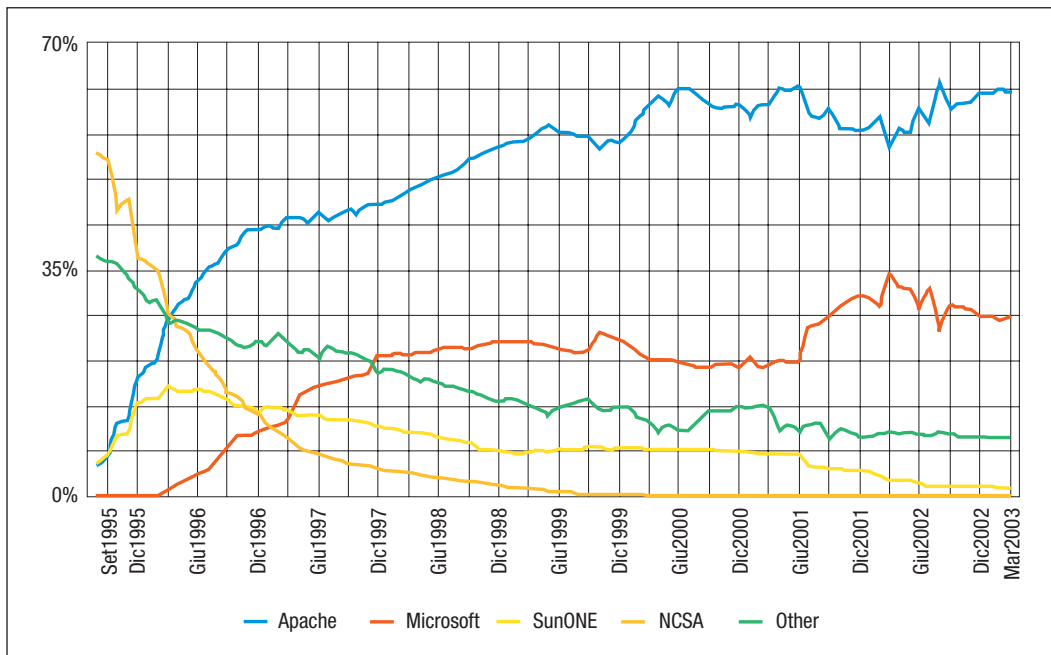


FIGURA 2
 Percentuale di server HTTP nella PAL italiana – fonte P.A.O.S.

e quindi, in termini strettamente rigorosi, non è interpretabile come una misura dell'adozione del software open source in generale all'interno di tali amministrazioni.

Secondo P.A.O.S. "Considerato però che i server Apache, con una frequenza di utilizzo a livello mondiale di oltre il 60% (secondo i dati forniti dalla rilevazione di Netcraft), rappresentano uno dei prodotti OS di più facile e ovvio utilizzo, argomentiamo che un'organizzazione che non utilizza neanche un server HTTP OS, mostra in media di possedere una scarsa propensione a utilizzare software OS in generale, e a maggior ragione verosimilmente non farà uso di applicativi OS di più difficile utilizzo, meno documentati e consolidati, rispetto alla famiglia dei server Apache". In questo senso, i dati della rilevazione, pur non riferendosi all'utilizzo del software OS in generale, forniscono un'indicazione utile per valutare la "propensione" dell'amministrazione pubblica all'utilizzo di software open source.

5. RUOLO DEL SOFTWARE OPEN SOURCE NELLA PUBBLICA AMMINISTRAZIONE

Nella PA la sfida dell'OSS si gioca sul *software applicativo custom* prodotto dalle amministrazioni pubbliche per far fronte a esigenze amministrative specifiche o, più in generale,

di amministrazione telematica, quali la gestione di strade o ospedali, l'istruzione, il pagamento d'imposte, la giustizia e la gestione del territorio [15].

Il software prodotto da o per le amministrazioni pubbliche non è costituito da "pacchetti standard" che possono essere utilizzati "tali e quali" da altri utenti.

La presunta assenza di finalità commerciali per quanto riguarda i canoni di licenza e la necessità di fornire il software con il codice, per consentire di adattarlo alle realtà locali prima di applicarlo e ridistribuirlo, conducono naturalmente ad aderire al modello OSS [6].

Fra i requisiti della PA vanno ricordati, fra gli altri, *economicità, indipendenza dai fornitori, sicurezza, riusabilità e interoperabilità*.

5.1. Benefici del modello open source nella pubblica amministrazione

L'adozione di OSS porta normalmente a un *risparmio iniziale* in termini di costi per licenze. Un confronto economico corretto deve essere però compiuto non solo sulla spesa iniziale, ma tra il TCO (*Total Cost of Ownership*) delle soluzioni open source e il TCO delle soluzioni proprietarie. Oltre al costo delle licenze, nel TCO confluiscono le spese dei servizi di supporto, della formazione, i costi di migrazione, d'installazione e di gestione. Le linee guida del governo inglese evidenziano molto prag-

maticamente il *value for money* come criterio di scelta per il software open source [13]. Un elemento a favore dell'OSS nella PA è l'*indipendenza dai fornitori*, consistente nel poter affidare il supporto di un prodotto open source a un'azienda scelta dal cliente, laddove nel mondo del software proprietario solo il produttore (o un suo partner autorizzato) può supportare il proprio software. Dispone, inoltre, del codice sorgente dei programmi utilizzati all'interno della propria organizzazione permette (anche se non garantisce) un grado maggiore di *sicurezza*. Sono, infatti, più agevoli i controlli interni (ove nei software proprietari ci si deve affidare ai produttori) alla ricerca di eventuali *back door* o debolezze sfruttabili da attacchi esterni. L'amministrazione dispone di un miglior controllo sulla politica di evoluzione del proprio parco applicativo e, in maniera generale, sul governo della gestione del patrimonio pubblico. Questo argomento assume una rilevanza particolare in aspetti connessi con la sicurezza, quali l'autenticazione e l'identificazione del cittadino o quelli relativi all'integrità, confidenzialità

I formati "aperti" e "standard" sono:

1. Un "formato aperto" può essere definito come la "modalità di rappresentazione dei dati in forma elettronica, deliberatamente resa pubblica, completamente documentata ed utilizzabile da chiunque". In questo senso, per esempio il formato utilizzato da "OpenOffice.org" è un formato aperto in quanto:
 - È una modalità di rappresentazione dei dati in forma elettronica.
 - È esaustivamente documentato e utilizzabile da chiunque: i dati vengono rappresentati nativamente in XML (*eXtensible Markup Language*) e salvati come documenti XML la cui struttura è definita in una DTD, grammatica di una classe di documenti XML, pubblica.
2. Un **formato è standard** quando è definito da un ente di standardizzazione (come nel caso di HTML) o è di fatto condiviso da una comunità (come nel caso di PDF, *Portable Document Format*).
3. Un **formato è uno standard aperto** quando soddisfa il requisito di pubblicità e di normazione (per esempio, XML e HTML sono standard aperti perché le loro specifiche sono pubblicamente documentate, definite e mantenute da un ente di standardizzazione, il W3C).
4. Un **formato testo** è un formato appartenente al sottoinsieme dei formati, caratterizzato dalla corrispondenza biunivoca fra un carattere alfanumerico ("c" minuscola) di un determinato insieme (latino, greco, cirillico, arabo, devanagari ecc.) e il valore del gruppo di bit costituenti l'unità di informazione di quel formato (p.e. 1 byte ASCII, 1-6 byte UTF-8 ecc.).
5. I **formati testo aperti standard** consentono l'indipendenza dall'evoluzione tecnologica; pertanto, le informazioni rappresentate con questo formato sono recuperabili anche molto tempo dopo la generazione, senza necessità di pesanti riconversioni. Questa caratteristica è ancor più vera per quei formati come SGML e XML che al dato associano la relativa descrizione (metadato) in linguaggio naturale: fra cinquanta anni, purché sia possibile leggere un *file* di testo con codifica ASCII o UNICODE, sarà sempre possibile recuperare i dati di un documento XML e capirne semantica e struttura.

e all'accessibilità dei dati nel corso del tempo. In generale, il software open source è più adatto a essere personalizzato o esteso come funzionalità rispetto a un software proprietario e quindi *riusato*.

Il ricorso al software libero può fungere da leva per la modernizzazione dei sistemi informatici dello Stato. La possibilità di ricorrere sia a software libero che proprietario aumenta le possibilità di scelta delle amministrazioni e consente [1]:

■ di accedere a un patrimonio considerevole di software spesso di qualità e conforme agli standard;

■ di governare il rapporto costo totale della soluzione/rispondenza ai bisogni attraverso il rafforzamento della concorrenza, allo scopo di mantenere questo rapporto al livello più basso possibile;

■ di governare il software e di avere la possibilità di assicurarne la perennità. In altre parole, l'amministrazione è messa in grado di capire e modificare il software per facilitare la sua integrazione e/o la sua evoluzione.

Cruciale per le Pubbliche Amministrazioni è l'interscambio di dati, da cui scaturisce la necessità di usare **formati aperti e standard**, scelta che assicura:

■ **Indipendenza**. La documentazione pubblica e completa del formato consente l'indipendenza da uno specifico prodotto e fornitore; tutti possono sviluppare applicazioni che gestiscono un formato aperto.

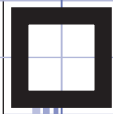
■ **Interoperabilità**. Usando formati aperti standard, sistemi eterogenei sono in grado di condividere gli stessi dati.

■ **Neutralità**. I formati aperti non obbligano a usare uno specifico prodotto, lasciando libero l'utente di scegliere sulla base del rapporto qualità/prezzo.

Inoltre, i *formati testo aperti standard* comportano l'ulteriore beneficio della persistenza, caratteristica importante per la tutela del patrimonio informativo nel tempo a fronte del mutamento tecnologico.

5.2. Limiti e rischi

Si noti che l'interoperabilità non comporta anche la portabilità, che invece è un punto a sfavore degli OSS. Le diverse distribuzioni di Linux, per esempio, contengono versioni differenti delle stesse librerie, per cui gli svilup-



patori hanno difficoltà nel garantire la portabilità delle loro applicazioni.

Sempre a sfavore si citano la bassa compatibilità con standard commerciali.

Infine, la principale difficoltà, come riconosciuto dalla stessa comunità internazionale di utenti Linux, è la mancanza di *driver*: la maggioranza dei produttori di periferiche non forniscono driver per Linux, dunque la lista dell'*hardware* compatibile è limitata ai dispositivi cui la comunità degli sviluppatori open source ha accesso. Quando viene lanciata sul mercato una nuova periferica, occorrono mesi prima che i driver siano disponibili, ammesso che i produttori forniscano le interfacce necessarie per lo sviluppo dei driver stessi; il problema è particolarmente evidente per le schede video e per i *modem*.

Qualsiasi sviluppo e messa in produzione di software comporta dei rischi. L'utilizzazione del software libero, lasciando un più grande margine di manovra agli utilizzatori, necessita anche da parte loro di una comprensione chiara delle nuove implicazioni legate al loro maggior controllo sul software; si enunciano di seguito i principali fattori di rischio [1]:

□ sviluppare utilizzando dei componenti di software libero le cui licenze sono fra loro incompatibili.

La disponibilità del codice sorgente permette tecnicamente di comporre un software utilizzando il codice sorgente dei componenti o dei software completi soggetti a licenze differenti. I responsabili di progetto devono vigilare sulla compatibilità dei componenti al fine di garantire la legalità del prodotto finale.

□ Implicazione dell'amministrazione, in quanto responsabile del software sviluppato o modificato, nella problematica del rispetto del diritto d'autore o della garanzia.

La diffusione pubblica del software realizzato o modificato dall'amministrazione rende necessario assicurarsi che i diritti degli autori siano rispettati e precisare le garanzie che vengono fornite con il software (sia libero che proprietario). La disponibilità del codice sorgente rende più facile la verifica di eventuali violazioni del diritto di autore e richiede, pertanto, una maggiore attenzione all'origine del codice sorgente incorporato nel software.

6. CONCLUSIONI

Un IT manager di una PA si domanderà dunque: "In che misura o in che condizioni il software sviluppato da un'amministrazione corrisponde al modello di sviluppo a codice sorgente aperto"?

Se si analizzano i processi di sviluppo, acquisizione, installazione del software e le risorse coinvolte (umane, organizzative, progettuali, finanziarie, distributive e d'informazione), proprie del nuovo modello di sviluppo OSS - definito *bazar*, in contrapposizione al modello tradizionale di costruzione della *cattedrale* [14] - si riscontra che esso si basa sui contributi di migliaia di programmatori che lavorano in modo totalmente volontario, scambiandosi idee e file attraverso Internet, e giungono in assenza di autorità centrale a realizzare programmi complessi.

In realtà, i progetti di successo (uno per tutti Linux) si sono aggregati intorno a figure carismatiche, come appunto Linus Torvalds. La constatazione che i progetti più riusciti e i software più affermati sviluppati nel paradigma OSS siano software di sistema (Sistemi Operativi, web server, *mail system*, linguaggi ecc.) spinge, inoltre, a pensare che tale modello abbia retto poiché di fatto la comunità spontanea degli sviluppatori condivideva le conoscenze dei requisiti e dell'architettura oggetto dello sviluppo [5].

In sintesi, è semplicistico pensare che nella PA sia possibile, senza sforzo iniziale e competenze interne, beneficiare di aggiornamenti e potenziamenti gratuiti, nel quadro di un sistema a codice sorgente aperto, a meno che non siano rispettate precise condizioni di sviluppo, quali [2, 6, 7]:

■ numero ragionevole di operatori "con lo stesso problema";

■ ripartizione iniziale e flessibile dei compiti di gestione/direzione tra persone diverse appartenenti a organizzazioni diverse;

■ documentazione esauriente;

■ niente codici monolitici;

■ software articolato in più segmenti di codice di dimensioni relativamente ridotte, così da agevolarne una gestione individuale;

■ chiara identificazione delle parti mature/stabili e di quelle che vanno ottimizzate in base al principio "dei rilasci frequenti".

D'altronde molte di queste considerazioni valgono anche per gli sviluppi proprietari. Forse il paradigma OSS può contribuire ad applicare buone regole troppo spesso solo enunciate e ad attivare un ciclo virtuoso di sviluppo del software, in un modello di *business* originale e promettente. La partita è aperta e, specialmente nella Pubblica Amministrazione, avvincente.

Bibliografia

- [1] ATICA (Agency for Information and Communications Technologies in the Civil Service): *Guide to choosing and using free software licences for government and public sector entities*. December 2002.
- [2] European Commission - DG Enterprise: *Study into the use of Open Source Software in the Public Sector An IDA Study - Interchange of Data between Administrations*. June 2001.
- [3] Faglioni G, et al: P.A.O.S - *Osservatorio Software Opensource nella Pubblica Amministrazione*. <http://ei.unibo.it/paos/>
- [4] Free Software Foundation <http://www.fsf.org>.
- [5] Fuggetta A: Open source software: an evaluation. *Journal of Systems and Software*, Vol. 66, Issue 1, 2003, p. 1-90.
- [6] Grasso F: Autorità per l'informatica nella pubblica amministrazione: *Il Software Open Source (OSS) - scenario e prospettive*, Giugno 2002 - n. 10 - supplemento al n. 3/2002 di Informazioni.
- [7] International Institute of Infonomics University of Maastricht, Berlecon Research GmbH Berlin: *FLOSS (Free/Libre and Open Source Software: Survey and Study)*. June 2002.
- [8] Meo A, Berra A: *Informatica solidale*. Bollati Boringhieri, 2001
- [9] Meo AR: Software libero e Open Source. *Mondo Digitale*, Anno I, n. 2, 2002, p.3-17.
- [10] Mockus A, et al: Two Case Studies of Open Source Software Development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, Vol. 11, n. 3, July 2002, p. 309-346.
- [11] Netcraft survey <http://www.netcraft.com/>
- [12] Open Source Initiative <http://www.opensource.org>.
- [13] *Open Source Software - USE Within UK Government*. Version 1 Date: 15/7/02 http://www.govtalk.gov.uk/documents/oss_policydocument_2002-07-15.doc.
- [14] Raymond E: *The Cathedral and the Bazaar*. Paperback edition, 2001.
- [15] Russo P, Sissa G: *Il Governo Elettronico*. Apogeo 2000.

GIOVANNA SISSA Ha partecipato ai lavori della "Commissione sul software a codice sorgente aperto nella PA" del Ministero per l'Innovazione e le Tecnologie. È esperta indipendente per la Commissione Europea - DG INFOS dal 1997.

Consulente per la ricerca e sviluppo nell'Industria e nella PA, ha diretto in passato lo sviluppo di sistemi esperiti per Ansaldo Trasporti e si è occupata di trasferimento tecnologico dell'ICT e di UNIX.

Presso l'Università di Genova, dove è stata professore a contratto di Intelligenza Artificiale, dirige dal 2001 l'Osservatorio Tecnologico del MIUR, un servizio di trasferimento tecnologico dell'ICT alle scuole. È coautrice del libro "Il Governo Elettronico".

Membro del gruppo italiano di definizione della certificazione ECDL Advanced - IT Administrator, ha contribuito alla stesura del volume "Uso avanzato delle reti".

giosissa@tin.it