



Sistema pubblico di cooperazione: ACCORDO DI SERVIZIO

Versione 1.1



DigitPA

INDICE

1.	MODIFICHE DOCUMENTO	3
2.	RIFERIMENTI	4
3.	OBIETTIVI E CONTESTO DI RIFERIMENTO	6
3.1.	Scopi del documento.....	7
3.2.	Note di lettura del documento	7
3.3.	Note sul Copyright	8
4.	CONCETTI DI BASE	9
5.	L'ACCORDO DI SERVIZIO (AS)	11
5.1.	Ciclo di Vita dell'Accordo di Servizio.....	11
5.2.	Struttura di un Accordo di Servizio.....	17
5.2.1.	<i>Parte Comune dell'AS</i>	<i>18</i>
5.2.2.	<i>Parte Specifica dell'AS.....</i>	<i>19</i>
5.2.3.	<i>Note Evolutive.....</i>	<i>19</i>
6.	SPECIFICA FUNZIONALE DEL SERVIZIO	22
6.1.	Specifica delle Interfacce	23
6.2.	Specifica delle Conversazioni.....	25
6.2.1.	<i>Modello per la Specifica delle Conversazioni</i>	<i>26</i>
6.2.1.1.	<i>Livello Concettuale</i>	<i>26</i>
6.2.1.2.	<i>Livello Logico</i>	<i>30</i>
6.2.2.	<i>Specifica delle Conversazioni nell'AS</i>	<i>30</i>
6.3.	Rappresentazione in XML	33
6.3.1.	<i>Raccomandazioni per l'utilizzo di WSDL.....</i>	<i>34</i>
6.3.2.	<i>WSBL – Web Service Behavior Language.....</i>	<i>37</i>
7.	SPECIFICA DEI LIVELLI DI SERVIZIO.....	42
7.1.	Concetti di Base	42
7.2.	Approccio Metodologico per SPCoop	46
8.	SPECIFICA DELLE CARATTERISTICHE DI SICUREZZA.....	51
9.	DOMINIO DI COOPERAZIONE, ACCORDO DI COOPERAZIONE E RELAZIONI CON L'ACCORDO DI SERVIZIO.....	54

1. MODIFICHE DOCUMENTO

Descrizione Modifica	Edizione	Data
Versione 1.0	1.0	14/10/2005
Adeguamento documentazione DigitPA	1.1	25/07/2011

2. RIFERIMENTI

Codice	Titolo
[Chessman&Daniels]	J. Cheesman, J. Daniels, "UML Components". Addison Wesley 2000
[IDABC]	Interoperable Delivery of European eGovernment Services to public Administrations, Business and Citizens. http://europa.eu.int/idabc/
[IJCIS2004]	B. Benatallah, F. Casati, H. Skogsrud, F. Toumani, "Abstracting and Enforcing Web Service Protocols", International Journal on Cooperative Information Systems, 13(4), 2004.
[IJCIS2005]	D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, M. Mecella, "Automatic Service Composition based on Behavioral Descriptions", International Journal on Cooperative Information Systems, 14(4), 2005
[QualitàDati]	P.D. Falorsi, M. Scannapieco (a cura di): Principi Guida per il Miglioramento della Qualità dei Dati Toponomastici nella Pubblica Amministrazione. Progetto Congiunto AIPA – ISTAT, Volume finale, Luglio 2004
[RFC2119]	http://www.ietf.org/rfc/rfc2119.txt
[SLA]	Web Service Level Agreements: http://www.research.ibm.com/wsla/ GGF, Web Services Agreement Specification (WS-Agreement): http://www.gridforum.org/Meetings/GGF11/Documents/draft-ggf-graap-agreement.pdf
[Stirling]	C. Stirling, "Modal and Temporal Logics for Processes", Banff Higher Order Workshop, Springer Verlag LNCS 1043, 1996. On-line: http://homepages.inf.ed.ac.uk/cps/banff.ps
[TAPAS]	Trusted and QoS-Aware Provision of Application Services. Progetto (numero IST-2001-34069) finanziato dalla Comunità Europea nel 5° programma IST. http://www.newcastle.research.ec.org/tapas/index.html
[UnderstandingWSDL]	P. Brittenham, F. Curbera, D. Ehnebuske, S. Graham et al, "Understanding WSDL in a UDDI Registry", September 2002. On-line: http://www-106.ibm.com/developerworks/webservices/library/ws-wsdl/
[WSBook]	G. Alonso, F. Casati, H. Kuno, V. Machiraju, "Web Services. Concepts, Architectures and Applications", Springer Verlag 2004

Codice	Titolo
[WSDL]	Web Services Description Language (WSDL) 1.1: http://www.w3.org/TR/2001/NOTE-wsd1-20010315 Web Services Description Language (WSDL) 2.0, W3C Working Draft: http://www.w3.org/TR/2005/WD-wsd120-primer-20050510/ , http://www.w3.org/TR/2005/WD-wsd120-20050510 , http://www.w3.org/TR/2005/WD-wsd120-adjuncts-20050510/
[WS-SecurityFramework]	Web Services Security Framework. http://www-106.ibm.com/developerworks/library/ws-secroad/
[WS-BPEL]	Business Process Execution Language for Web Services: http://www-128.ibm.com/developerworks/library/specification/ws-bpel/
[WS-CDL]	Web Service Choreography Description Language: http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/

3. OBIETTIVI E CONTESTO DI RIFERIMENTO

Il quadro tecnico di riferimento per attuare la cooperazione applicativa tra le amministrazioni pubbliche nell'ambito del Sistema Pubblico di Connettività e Cooperazione è stato definito con l'approvazione, avvenuta nell'ottobre del 2004 da parte delle associazioni dei fornitori, delle amministrazioni partecipanti alla loro stesura e del Tavolo Congiunto Permanente della Conferenza Unificata Stato Regioni Città e Autonomie Locali, dei documenti che ne delineano l'architettura, l'organizzazione e le tecnologie standard da adottare.

Tali documenti hanno definito il "giusto" livello di condivisione che consente sia la maggiore stabilità nel tempo del modello rispetto al contesto organizzativo e tecnologico di riferimento, sia i necessari gradi di libertà per la sua implementazione. Il decreto legislativo n.42 del 28 febbraio 2005 che istituisce il Sistema Pubblico di Connettività e Cooperazione, ne stabilisce i valori fondanti, la validità giuridica, nonché il modello di governo strategico ed operativo ed i ruoli del CNIPA e delle Regioni in tali ambiti.

I suddetti documenti tracciano un primo quadro di evoluzione del modello e definiscono gli ulteriori documenti di maggiore dettaglio da produrre per l'implementazione dei servizi previsti. La redazione di questi ultimi, come concordato, è stata portata avanti dal CNIPA ed ha dato luogo ai documenti di cui alla seguente tabella 1. Quest'ultimo insieme di documenti rappresenta le specifiche per la realizzazione e gestione dei servizi di cooperazione SPC e delle procedure di qualificazione, come già definito nei documenti approvati.

	Titolo Documento	Stato e Data Pubblicazione
1.	<i>Sistema Pubblico di Cooperazione:</i> <i>QUADRO TECNICO D'INSIEME</i>	Pubblicato V. 1.1 del 25/07/2011
2.	<i>Sistema Pubblico di Cooperazione:</i> <i>TERMINI E DEFINIZIONI</i>	Pubblicato V. 1.1 del 25/07/2011
3.	<i>Sistema Pubblico di Cooperazione:</i> <i>ACCORDO DI SERVIZIO</i>	Pubblicato V. 1.1 del 25/07/2011
4.	<i>Sistema Pubblico di Cooperazione:</i> <i>PORTA DI DOMINIO</i>	Pubblicato V. 1.1 del 25/07/2011
5.	<i>Sistema Pubblico di Cooperazione:</i> <i>BUSTA DI E-GOV</i>	Pubblicato V. 1.2 del 25/07/2011
6.	<i>Sistema Pubblico di Cooperazione:</i> <i>SERVIZI DI REGISTRO</i>	Pubblicato V. 1.1 del 25/07/2011
7.	<i>Sistema Pubblico di Cooperazione:</i> <i>SERVIZI DI SICUREZZA</i>	Pubblicato V. 1.1 del 25/07/2011
8.	<i>Sistema Pubblico di Cooperazione:</i> <i>CONVENZIONI DI NOMENCLATURA E SEMANTICA</i>	Pubblicato V. 1.1 del 25/07/2011
9.	<i>Sistema Pubblico di Cooperazione:</i> <i>ESERCIZIO E GESTIONE</i>	Pubblicato V. 1.1 del 25/07/2011

Tabella 1. Documenti di specifica del SPCoop

3.1. Scopi del documento

Il presente documento specifica la struttura dell'Accordo di Servizio, definisce le informazioni che in esso devono essere formalizzate e le modalità di tale formalizzazione.

La redazione è stata ad opera di:

- Roberto Baldoni (Università di Roma “La Sapienza”);
- Stefano Fuligni (CNIPA);
- Massimo Mecella (Università di Roma “La Sapienza”);
- Francesco Tortorelli (CNIPA).

3.2. Note di lettura del documento

Nella definizione dei requisiti, delle specifiche e delle regole descritte nei documenti precedentemente indicati sono utilizzate le parole chiave DEVE, NON DEVE, OBBLIGATORIO, VIETATO, DOVREBBE, CONSIGLIATO, NON DOVREBBE, SCONSIGLIATO, POTREBBE, OPZIONALE che devono essere interpretate in conformità con [RFC2119]. In particolare:

- DEVE, OBBLIGATORIO significano che la definizione è un requisito assoluto, la specifica deve essere implementata, la consegna è inderogabile.
- DOVREBBE, CONSIGLIATO significano che in particolari circostanze possono esistere validi motivi per ignorare un requisito, non implementare una specifica, derogare alla consegna, ma che occorre esaminare e valutare con attenzione le implicazioni correlate alla scelta.
- PUÒ, OPZIONALE significano che un elemento della specifica è a implementazione facoltativa.
- NON DOVREBBE, SCONSIGLIATO significano che in particolari circostanze possono esistere validi di motivi per cui un elemento di specifica è accettabile o persino utile, ma, prima di implementarlo, le implicazioni correlate dovrebbero essere esaminate e valutate con attenzione.
- NON DEVE, VIETATO significano che c'è proibizione assoluta di implementazione di un determinato elemento di specifica.

3.3. Note sul Copyright

Il presente documento ed i suoi contenuti sono di proprietà del Centro nazionale per l'informatica nella pubblica amministrazione (CNIPA) e sono protetti dalle norme sul diritto d'autore e dalle altre norme applicabili.

Il presente documento ed i suoi contenuti sono messi a disposizione sulla base dei termini della licenza d'uso disponibile al seguente indirizzo:

http://www.digitpa.gov.it/sites/default/files/allegati_tec/Licenza_duso_documenti_SPCoop_v.2.0.pdf

4. CONCETTI DI BASE

In questa sezione sono riportati e riassunti per completezza alcuni concetti introdotti e sviluppati nei documenti SPCoop della fase due, a cui il lettore è invitato a riferirsi ogniqualvolta sia necessario approfondire specifici dettagli.

Nell'architettura SPCoop, tra due sistemi s'instaura una relazione di servizio se uno dei due sistemi, che riveste il ruolo di *fruitore*, utilizza i risultati di trattamenti informatici effettuati dall'altro sistema, che riveste il ruolo di *erogatore*. Intuitivamente, un *servizio* è l'insieme dei risultati prodotti dai trattamenti effettuati dal sistema erogatore e utilizzati dai trattamenti effettuati dal sistema fruitore; più specificatamente l'insieme dei risultati prodotti è *l'insieme delle prestazioni di servizio*; tale insieme delle prestazioni di servizio è organizzato in *classi* (tipi) di prestazioni, che sono le *funzionalità di servizio*. La descrizione di un servizio è la descrizione delle classi di prestazioni del servizio.

Nell'architettura SPCoop, la sola ed esclusiva modalità d'interazione tra il sistema erogatore e il sistema fruitore di un servizio è lo scambio di messaggi, sulla base degli standard tecnologici Web Service. Lo scambio di messaggi è utilizzato per coordinare l'erogazione/fruizione delle prestazioni di servizio e, eventualmente, per trasportare dati la cui fornitura fa parte di dette prestazioni. In particolare, sono prescritti tre tipi di scambio elementare:

- *messaggio senza replica*: un sistema *mittente* invia un messaggio a un sistema *destinatario*;
- *messaggio/replica sincroni*: un sistema detto *richiedente* trasmette un *messaggio sincrono* a un sistema detto *rispondente* e si mette in attesa della *replica sincrona*. Successivamente il rispondente trasmette una *replica sincrona* al richiedente correlata logicamente con il messaggio;
- *messaggio/replica asincroni*: un sistema detto *richiedente* trasmette un *messaggio asincrono* a un sistema detto *rispondente*. Il rispondente trasmette in seguito una *replica asincrona* al richiedente correlata logicamente con il messaggio ricevuto. La correlazione a livello logico tra messaggio e replica asincroni è rappresentata esplicitamente dall'inserzione nella replica di un identificatore di correlazione.

Lo scambio elementare di messaggi è utilizzato per coordinare l'erogazione/fruizione delle prestazioni di servizio, ovvero gli *scenari di coordinamento* della prestazioni di servizio. Nell'architettura SPCoop qualsiasi scenario di coordinamento di una prestazione di servizio risulta esclusivamente della combinazione di scambi elementari di tipo messaggio senza replica, messaggio/replica sincroni, messaggio/replica asincroni. Sono previsti alcuni tipi di *scenari elementari di coordinamento* dell'erogazione/fruizione del servizio:

- *Richiesta senza risposta*: il fruitore trasmette all'erogatore un messaggio contenente una richiesta di prestazione e continua il trattamento. Tale scenario può essere implementato utilizzando come tipo di interazione il *messaggio senza replica*.
- *Richiesta/risposta*: il fruitore trasmette all'erogatore un messaggio contenente una richiesta di prestazione di servizio. L'erogatore esegue il trattamento di erogazione della prestazione e trasmette un messaggio di risposta contenente il resoconto di erogazione e

eventualmente i dati la cui fornitura è parte della prestazione. Le modalità dell'interazione (dal punto di vista del fruitore) che implementano lo scenario di richiesta/risposta possono essere sia sincrone che asincrone:

- *Richiesta/risposta sincrone*: il sistema fruitore prepara e emette il messaggio di richiesta della prestazione e blocca la sua linea di esecuzione in attesa della risposta. Alla ricezione della risposta la linea di esecuzione del fruitore, riprende l'esecuzione. Lo scenario richiesta/risposta sincrone può essere implementato direttamente utilizzando uno scambio elementare di tipo *messaggio/replica sincrónico*.
- *Richiesta/risposta asincrone*: il fruitore del servizio prepara la richiesta di prestazione, la trasmette all'erogatore e continua l'esecuzione. La richiesta viene trasmessa al sistema erogatore che esegue i trattamenti, prepara e emette la risposta. La ricezione della risposta ha come effetto presso il fruitore l'avvio di una linea di esecuzione indipendente che esegue i trattamenti conseguenti alla ricezione della risposta. Può essere implementato utilizzando lo scambio elementare di tipo *messaggio/replica asincrono*. L'identificatore di correlazione messaggio/replica può essere utilizzato come identificatore di correlazione richiesta/risposta.
- *Notificazione senza risposta*: l'erogatore del servizio trasmette di sua iniziativa un messaggio di notificazione al fruitore (che può contenere il resoconto di un trattamento effettuato o la *notifica di un evento*). Tale scenario può essere implementato usando uno scambio elementare di tipo *messaggio senza replica*.
- *Notificazione/risposta*: l'erogatore trasmette un messaggio di notificazione e successivamente il fruitore trasmette un messaggio di risposta. Può essere implementata con modalità sincrona o asincrona e quindi essere implementato rispettivamente dallo scambio elementare messaggio/replica sincrónico o asincrono.

5. L'ACCORDO DI SERVIZIO (AS)

Per instaurare una relazione di servizio tra sistemi è obbligatorio definire un accordo esplicito sulla erogazione/fruizione delle prestazioni del servizio: l'Accordo di Servizio (AS). Esso definisce le prestazioni del servizio e le modalità di erogazione/fruizione, ovvero le *funzionalità* del servizio, le *interfacce* di scambio dei messaggi tra erogatore e fruitore, i *requisiti di qualità di servizio* dell'erogazione/fruizione, ed i *requisiti di sicurezza* dell'erogazione/fruizione. Inoltre mantiene un riferimento all'ontologia/schema concettuale che definisce la semantica dell'informazione veicolata dal servizio.

I termini dell'AS sono consegnati in un insieme di documenti, secondo la struttura che verrà dettagliata nel proseguo di questo documento. Detto insieme di documenti costituisce:

- una specifica per l'implementazione dei sistemi erogatore e fruitore: è utilizzato dai progettisti e dagli ambienti di sviluppo nelle fasi di implementazione dei sistemi erogatore e fruitore;
- un riferimento per l'esecuzione e la gestione dei sistemi erogatore e fruitore: è utilizzato dai sistemi di gestione dell'esecuzione durante l'operatività dei sistemi.

5.1. Ciclo di Vita dell'Accordo di Servizio

La definizione di un servizio è il processo organizzativo il cui prodotto è l'Accordo di Servizio (AS). Un AS fa riferimento alla coppia <erogatore, fruitore> delle organizzazioni erogatrice e fruitrice che lo hanno sottoscritto: ogni AS è necessariamente caratterizzato dalla coppia <erogatore, fruitore>, e date due organizzazioni che cooperano su SPCoop, una come erogatore e l'altra come fruitore di un particolare servizio, necessariamente è definito un AS relativo ad esse. Proprio in base a questa proprietà, quando necessario nel resto del documento, si userà la notazione AS<erogatore, fruitore> ogniqualvolta sia necessario esplicitamente indicare la coppia che ha sottoscritto l'accordo di servizio.

Un accordo di servizio è composto di una *parte comune*, che formalizza gli aspetti riusabili in differenti contesti, e da una *parte specifica*, che precisa e dettaglia la parte comune istanziandola sulla particolare coppia <erogatore, fruitore> a cui l'accordo di servizio si riferisce. Ad esempio, nella parte specifica sono dettagliati i livelli di servizio ed i livelli di utilizzo data una particolare coppia <erogatore, fruitore> che utilizza una parte comune come base per la definizione del proprio accordo. Verranno utilizzati i termini (i) *derivare*, per indicare il fatto che la parte comune è la base per la definizione di un accordo di servizio, e (ii) *specializzare* per indicare che una parte specifica dettaglia una comune. Valgono le seguenti relazioni: data una parte comune ed una parte specifica che la specializza alla coppia <erogatore, fruitore>, l'AS<erogatore, fruitore> che ne deriva è dato dall'unione di parte comune e parte specifica: (ovvero $AS\langle erogatore, fruitore \rangle = parte\ comune \cup parte\ specifica\langle erogatore, fruitore \rangle$).

La divisione dell'accordo di servizio in parte comune e parte specifica è funzionale all'utilizzo in scenari con più di due soggetti: infatti molte caratteristiche di un accordo di servizio sono spesso uguali e ripetute in casi di più erogatori e/o più fruitori, e quindi è da preferirsi una modularità nella specifica che metta a fattor comune tali parti. Poiché però alcune caratteristiche di un accordo di servizio dipendono strettamente dalla particolare coppia <erogatore, fruitore>, è sembrato naturale aggiungere alla parte comune una parte specifica, e considerare l'accordo di servizio come composizione delle due.

I servizi che sono disponibili su SPCoop si distinguono, dal punto di vista della portata della loro erogazione/fruizione, in 4 classi:

- *Servizi mono-erogatore/mono-fruitori*

Il servizio è erogato da un solo sistema erogatore indipendente S_{erog} , di cui è titolare un soggetto della comunità SPCoop, per la fruizione da parte di un unico altro sistema indipendente S_{fruit} di cui il titolare è un altro soggetto della comunità. Il titolare del sistema erogatore ha la responsabilità di gestione del ciclo di vita dell'accordo di servizio (indipendentemente dalle modalità del processo di definizione che ha condotto all'accordo) e dell'erogazione del servizio in conformità con l'accordo. Il titolare del sistema fruitore è responsabile della fruizione del servizio nei termini dell'accordo. In questo scenario, esiste una parte comune, specializzata da una sola parte specifica, derivando quindi il solo AS $\langle S_{erog}, S_{fruit} \rangle$, come mostrato in Figura 1.



Figura 1. Parte comune e parte specifica nel caso mono-erogatore/mono-fruitori

- *Servizi mono-erogatore/multi-fruitori*

Il servizio è erogato da un solo sistema erogatore indipendente S_{erog} , di cui è titolare un soggetto della comunità SPCoop, e destinato alla fruizione di una classe di M sistemi indipendenti (i fruitori) $S_{fruit_1}, \dots, S_{fruit_M}$, di cui sono responsabili altri soggetti della comunità SPCoop. Il titolare del sistema erogatore ha la responsabilità di gestione del ciclo di vita degli accordi di servizio (indipendentemente dalle modalità del processo che ha condotto agli accordi) e dell'erogazione del servizio in conformità con gli accordi. Ogni titolare di sistema fruitore è responsabile della fruizione del servizio nel rispetto dei termini del suo accordo di servizio con il titolare del sistema erogatore. In questo caso, esiste una sola parte comune da cui vengono derivati M AS^j ($j = 1 \dots M$), uno per ogni fruitore, come mostrato in Figura 2.

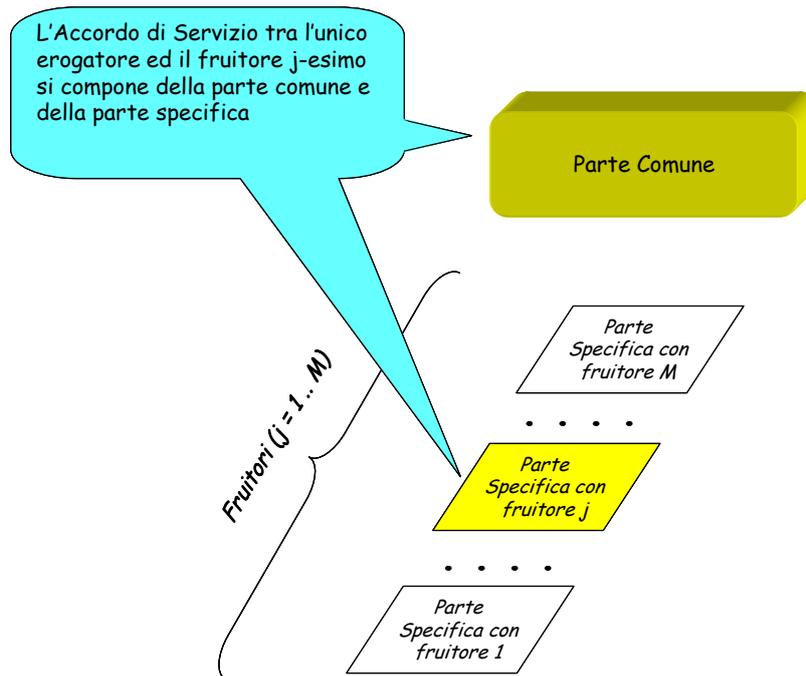


Figura 2. Parte comune e parti specifiche nel caso mono-erogatore/multi-fruitore

- *Servizi multi-erogatore/mono-fruitore*

Il servizio è erogato in conformità con la parte comune da un insieme di N sistemi erogatori indipendenti $S_{erog_1}, \dots, S_{erog_N}$, di cui sono titolari più soggetti differenti. Il servizio è destinato alla fruizione da parte di un solo sistema S_{fruit} di cui il titolare è un altro soggetto della comunità SPCoop. Il ciclo di vita dell'accordo di servizio è gestito da un soggetto, eventualmente terzo rispetto agli erogatori ed al fruitore, a cui viene delegato il compito (indipendentemente dal processo di definizione dell'accordo). Ciascun titolare di un sistema erogatore è responsabile dell'erogazione del servizio in conformità con il proprio accordo di servizio con il fruitore; il fruitore è responsabile della fruizione del servizio. In questo caso, esiste una sola parte comune da cui vengono derivati N ASⁱ ($i = 1 \dots N$), uno per ogni erogatore, come mostrato in Figura 3.

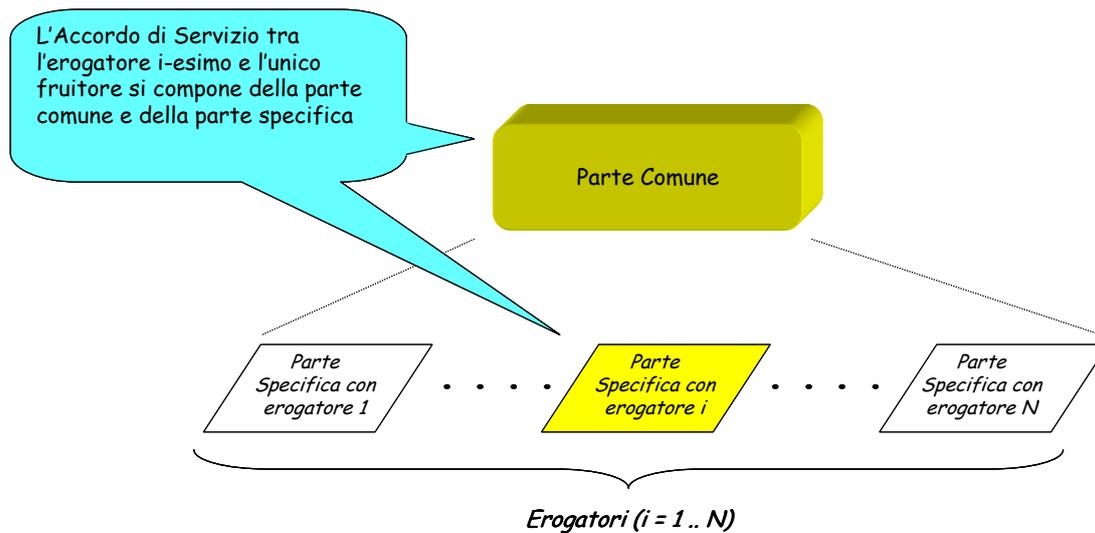


Figura 3. Parte comune e parti specifiche nel caso multi-erogatore/mono-fruitore

- *Servizi multi-erogatore/ multi-fruitore*

Il servizio è erogato da un insieme di N sistemi erogatori indipendenti $S_{erog_1}, \dots, S_{erog_N}$, i cui titolari sono soggetti diversi. I servizi (tutti conformi alla stessa parte comune) sono destinati alla fruizione di una classe di M sistemi fruitori indipendenti $S_{fruit_1}, \dots, S_{fruit_M}$ i cui titolari sono altri soggetti di SPCoop. Il ciclo di vita della parte comune è gestito da un soggetto, eventualmente terzo rispetto agli erogatori ed ai fruitori, a cui viene delegato il compito. Ogni erogatore è responsabile dell'erogazione del proprio servizio in conformità alla parte comune (che in generale non gestisce direttamente) ed alla parte specifica con il singolo fruitore (che invece gestisce direttamente). Ogni fruitore è responsabile della fruizione del servizio rispettando i termini della parte comune e delle clausole specifiche dell'accordo di servizio con il singolo erogatore. In questo caso, esiste una sola parte comune da cui vengono derivati $N \times M$ AS^{ij} ($i = 1.. N, j = 1 .. M$), uno per ogni combinazione erogatore/fruitore, come mostrato in Figura 4.

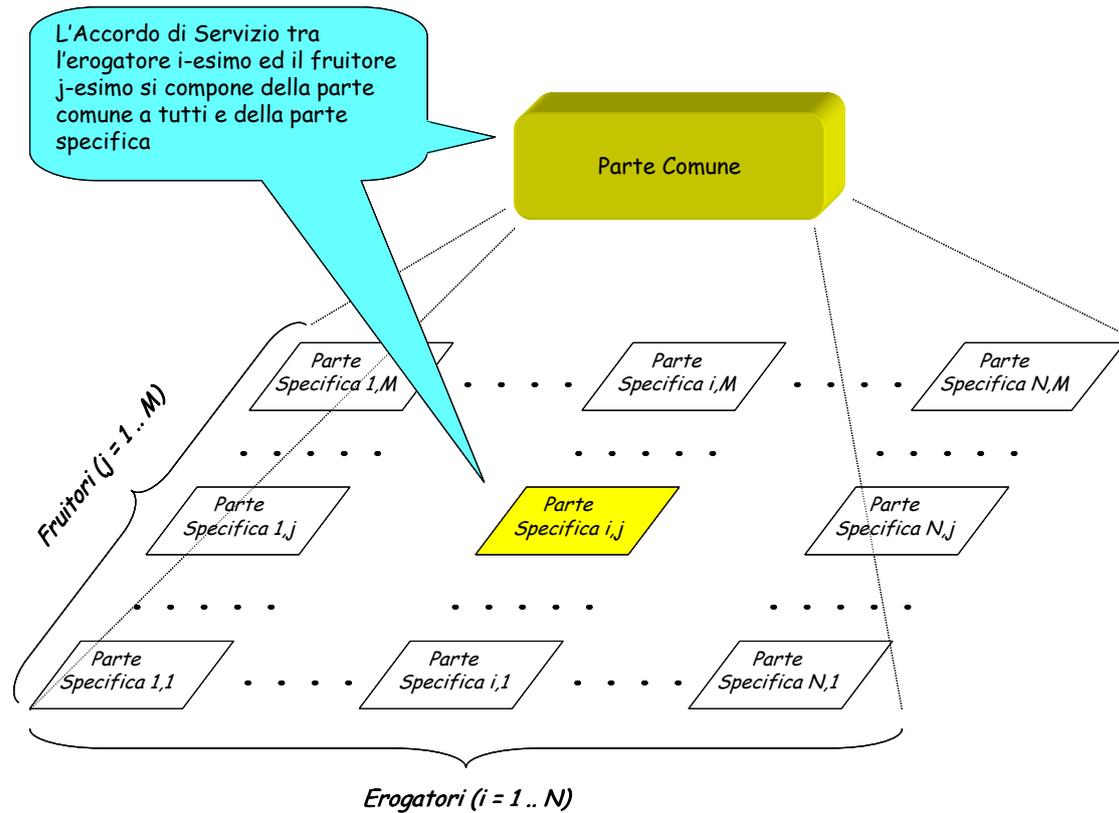


Figura 4. Parte comune e parti specifiche nel caso multi-erogatore/multi-fruitore

Si osservi che la divisione in parte comune e parte specifica è applicabile anche al caso, che si ritiene significativo (soprattutto in una fase iniziale di SPCoop), di definizione *unilaterale* di un accordo di servizio da parte di un singolo erogatore: in questo scenario (che può applicarsi sia alla classe mono fruitore – Figura 1 – che a quella multi fruitore – Figura 2), il soggetto erogatore definisce la parte comune e quello che si potrebbe indicare come un “template” della parte specifica, ovvero una parte specifica completa di tutte le caratteristiche (non negoziate con il possibile fruitore, ma “imposte”), senza però esplicita indicazione del fruitore; la parte specifica viene effettivamente istanziata sulla base del “template” nel momento stesso in cui il fruitore decide di accettarla e farla propria per poter utilizzare il servizio.

Si osservi invece che la definizione unilaterale della parte comune non preclude affatto una negoziazione con il/i fruitore/i della parte specifica e delle caratteristiche in essa specificate.

L'accordo di servizio tra un erogatore ed un fruitore è il cardine del ciclo di vita del servizio tra di essi. I servizi disponibili su SPCoop sono obbligatoriamente gestiti per versioni dell'accordo di servizio. Più versioni di uno stesso servizio (corrispondenti a versioni diverse dell'accordo di servizio) possono essere erogate/fruite nello stesso momento. Ogni versione segue un ciclo di vita autonomo. Si osservi che necessariamente versioni diverse dell'accordo di servizio tra un erogatore ed un fruitore differiscono nella parte comune.

Per ogni versione, il ciclo di vita è composto da 6 fasi:

Fase 1. Definizione dell'accordo di servizio.

- Fase 2. Registrazione dell'accordo di servizio sui Servizi di Registro SICA.
- Fase 3. Implementazione del servizio.
- Fase 4. Presentazione del servizio su SPCoop.
- Fase 5. Erogazione/fruizione del servizio su SPCoop.
- Fase 6. Dismissione dell'accordo di servizio e del servizio da SPCoop.

Le fasi del ciclo di vita del servizio sono descritte nella Tabella 2.

Fase	Descrizione
Fase 1. Definizione dell'accordo di servizio	Nella fase di definizione del servizio, il servizio viene ideato e formalizzato nella parte comune e nella parte specifica (eventualmente più di una). Il processo che porta alla definizione del servizio segue fondamentalmente due approcci: <ul style="list-style-type: none"> • <i>Approccio unilaterale.</i> Nell'approccio unilaterale, la parte comune viene concepita unilateralmente dal soggetto erogatore, o da un soggetto delegato. Le parti specifiche possono seguire l'approccio unilaterale o l'approccio concordato (vedi il prossimo paragrafo). • <i>Approccio concordato.</i> Nell'approccio concordato, l'accordo di servizio (sia nella parte comune e/o in quella specifica) è ideato in modo congiunto dall'erogatore (o dal suo rappresentante) e dal fruitore (o dal suo rappresentante).
Fase 2. Registrazione dell'accordo di servizio sui Servizi di Registro SICA.	In SPCoop, la registrazione dell'accordo di servizio è obbligatoria. La registrazione comprende molteplici attività da effettuare presso i Servizi di Registro SICA: <ul style="list-style-type: none"> • registrazione dei titolari dei sistemi erogatore e fruitore, • registrazione della parte comune, • registrazione della/e parte/i specifica/che, che in particolare include la registrazione dei porti di accesso ai sistemi erogatore e fruitore.
Fase 3. Implementazione del servizio.	Nella fase di implementazione, i titolari dei sistemi erogatori e fruitori implementano rispettivamente l'erogazione e la fruizione del servizio in conformità con l'accordo di servizio.
Fase 4. Presentazione del servizio su SPCoop.	Nella fase di presentazione del servizio, i titolari dei sistemi erogatori e fruitori presentano i componenti software che realizzano il servizio sulla rete, ovvero li mettono in esercizio. La presentazione è la fase che precede immediatamente la fase di erogazione/fruizione.
Fase 5. Erogazione/fruizione del servizio su SPCoop.	Nella fase di erogazione/fruizione il servizio è erogato in conformità con l'accordo di servizio. La fase di erogazione/fruizione comprende attività di tracciatura e può comprendere attività di monitoraggio definite nell'accordo di servizio e effettuate dai soggetti designati da detto accordo.

Fase	Descrizione
Fase 6. Dismissione dell'accordo di servizio e del servizio.	<p>La fase di dismissione (di una versione) del servizio prevede l'archiviazione (della versione) del servizio e dell'accordo di servizio e dei giornali di tracciatura e può comprendere l'archiviazione di eventuali supporti di non repudiabilità definiti nell'accordo di servizio.</p> <p>La fase di dismissione prevede la sottrazione (della versione) del servizio da SPCoop, dopo annuncio preventivo ai titolari dei sistemi fruitori.</p>

Tabella 2. Fasi del ciclo di vita del servizio.

5.2. Struttura di un Accordo di Servizio

Se da un lato l'accordo di servizio si scompone nella parte comune e nella parte specifica, dall'altro in esso si possono distinguere una *componente formalizzata* della specifica ed una *componente non formalizzata*:

- Specifica Formalizzata.* Contiene la specifica delle interfacce del servizio¹, delle conversazioni supportate dal servizio, dei livelli di qualità del servizio e delle caratteristiche di sicurezza. Specifica formalizzata significa che essa è espressa attraverso delle opportune sintassi basate su XML; a seconda dei casi, tali sintassi XML potranno essere: (i) standard riconosciuti (ad es., WSDL – Web Service Description Language, versione 1.1), (ii) proposte particolarmente significative da parte del mondo della ricerca o dell'industria non ancora standardizzate, ma ritenute comunque sufficientemente mature (ad es., WSLA – Web Service Level Agreement, versione 1.0, WS-Agreement, versione 1.0, ecc.), (iii) proposte specificatamente definite per SPCoop (ad es., WSBL – Web Service Behavior Language, versione 1.0); nel caso di sintassi XML appositamente definite per SPCoop, nel presente documento verrà riportata sia la definizione esatta della sintassi (XML Schema) che la semantica dei costrutti.

In ogni caso, indipendentemente dalla stato di standardizzazione della sintassi XML, il suo utilizzo per la specifica di una parte dell'AS rende “formale” tale specifica.
- Specifica Non-formalizzata.* Contiene la specifica in modo rigoroso ma semi-formale delle interfacce del servizio, delle conversazioni supportate dal servizio, dei livelli di qualità del servizio e dei requisiti di sicurezza. Si tratta cioè dello stesso contenuto informativo della componente formalizzata, solamente espresso in modo differente. Questa componente può contenere differenti documenti, che non vengono qui normati: file HTML che descrivono in linguaggio naturale il funzionamento del servizio, diagrammi UML che descrivono l'interfaccia del servizio (ad es., class diagram) e/o le conversazioni supportate (ad es., state diagram, sequence diagram), ecc.

Si ha cioè che lo stesso contenuto informativo dell'Accordo di Servizio può essere espresso sia in modo formalizzato attraverso opportuni linguaggi XML, oppure in modo

¹ Come verrà descritto nel seguito, nella specifica delle interfacce deve essere considerato anche un riferimento ad un eventuale documento che descriva la semantica delle informazioni veicolate dal servizio (attraverso un'ontologia/schema concettuale).

rigoroso ma non formale. In Figura 5 sono mostrate le due coordinate (parte comune vs. parte specifica, componente formalizzata vs. componente non formalizzata) in cui è possibile sezionare un accordo di servizio.

5.2.1. Parte Comune dell'AS

La parte comune dell'AS contiene la Specifica dell'Interfaccia, la Specifica delle Conversazioni ed il Riferimento alla Semantica dell'Informazione Veicolata.

- La **Specifica dell'Interfaccia** (come verrà dettagliato in § 6.1) si compone di 3 specifiche WSDL (Web Service Description Language, versione 1.1) *astratte*: (i) descrizione dell'interfaccia del servizio a livello di scenario di coordinamento (concettuale), (ii) descrizione della parte di competenza del fruitore a livello di scambio elementare di messaggi (logico), (iii) descrizione della parte di competenza dell'erogatore a livello di scambio elementare di messaggi (logico).

WSDL astratto, così come dettagliato in [UnderstandingWSDL], significa che questi file WSDL contengono i soli costrutti `types`, `message`, `operation` e `port type`, in quanto non specificano aspetti legati alla particolare implementazione del servizio (ad es., il binding o la URI di accesso).

Le dizioni “concettuale” e “logico” con cui vengono etichettati i file WSDL sono utilizzate in analogia a quanto viene proposto nelle comuni metodologie per il progetto delle basi di dati. A livello concettuale, viene descritta l'interfaccia del servizio indipendentemente dallo scambio elementare di messaggi su cui poi effettivamente il servizio verrà realizzato. Il livello logico descrive come l'interfaccia del servizio viene effettivamente realizzata utilizzando particolari scambi elementari di messaggi. Dato il modello proposto in SPCoop (riportato in § 4), a questo livello sarà presente un file che descrive il Web Service lato fruitore ed il Web Service lato erogatore. Questa coppia di Web Service insieme sono la realizzazione del servizio. Infine ci sarà un livello fisico in cui effettivamente verranno dettagliati i binding e le URI di accesso di questi due Web Service. Questo non è però incluso nella parte comune, bensì nella parte specifica (in quanto dipendente dalla particolare coppia <erogatore, fruitore>).

- La **Specifica delle Conversazioni** descrive le conversazioni supportate dal servizio; per conversazione si intende una sequenza di operazioni/scambi di messaggi che il servizio effettivamente supporta. Tutte le possibili conversazioni sono descritte in modo compatto attraverso una Macchina a Stati Finiti (FSM – Finite State Machine) che viene rappresentata attraverso una sintassi XML appositamente definita, WSBL (Web Service Behavior Language, versione 1.0, introdotto in dettaglio in § 6.3). Così come per la specifica dell'interfaccia, anche la specifica delle conversazioni consiste di 3 specifiche WSBL: (i) quella a livello concettuale, (ii) quella a livello logico dell'erogatore e (iii) quella a livello logico del fruitore. In § 6.2 verrà descritta in dettaglio la specifica delle conversazioni.
- Il **Riferimento alla Semantica dell'Informazione Veicolata** contiene un riferimento all'ontologia/schema concettuale (o porzioni di essa) che definisce la semantica dell'informazione veicolata dal servizio. Come descritto nel documento QUADRO TECNICO D'INSIEME, accanto agli Accordi di Servizio, è presente, nel modello di cooperazione, la nozione di semantica sottostante alle informazioni che attraverso un

servizio vengono veicolate, e tale semantica è rappresentata in un apposito catalogo attraverso ontologie/schemi concettuale. Un AS definisce allora un riferimento a tale ontologia/schema concettuale. Per motivazioni differenti, discusse nel suddetto documento, si è scelto di non aggregare tale descrizione della semantica del servizio nel suo AS, ma di lasciarlo come elemento esterno, semplicemente riferito dall'AS².

•

5.2.2. Parte Specifica dell'AS

La parte specifica dell'AS contiene la Specifica dei Porti di Accesso, la Specifica dei Livelli di Servizio e la Specifica delle Caratteristiche di Sicurezza.

- La **Specifica dei Porti di Accesso** descrive gli aspetti implementativi (livello fisico) dei Web Service dell'erogatore e fruitore, ovvero l'URI dei punti d'accesso ed i binding (basati sulla Busta eGov) delle varie operazioni.
- La **Specifica dei Livelli di Servizio** descrive gli elementi necessari per poter definire i livelli di servizio, e le obbligazioni che legano la specifica coppia <erogatore, fruitore> con le azioni correlate. In § 7 si considererà in maggior dettaglio questo elemento.
- La **Specifica delle Caratteristiche di Sicurezza** descrive le caratteristiche di sicurezza supportate e richieste dal servizio. In § 8 si considererà in maggior dettaglio questo elemento.

5.2.3. Note Evolutive

In questa versione delle specifiche dell'AS, solamente le parti di (i) Specifica dell'Interfaccia (ovvero i tre documenti WSDL) e di (ii) Specifica dei Porti di Accesso sono obbligatorie; le altre parti sono tutte *transitoriamente* opzionali; è comunque fortemente consigliata la loro inclusione nell'AS.

L'opzionalità di queste quattro parti, ovvero di: (iii) Specifica delle Conversazioni, (iv) Riferimento alla Semantica dell'Informazione Veicolata, (v) Specifica dei Livelli di Servizio e (vi) Specifica delle Caratteristiche di Sicurezza, è dovuta a considerazioni di pragmatica: soprattutto nella fase iniziale di SPCoop, infatti, non è garantita la presenza di adeguate risorse e strumenti di supporto alle Amministrazioni per la loro stesura. È, però, prevedibile che nel tempo, al crescere dei servizi esposti e dei rispettivi AS pubblicati, le singole Amministrazioni prendano confidenza con SPCoop e con le relative specifiche e che, quindi, si verifichi un lento ma progressivo diffondersi di una cultura "orientata ai servizi". Il piano evolutivo delle specifiche prevede che, progressivamente, anche le altre componenti vengano rese obbligatorie nella loro veste formalizzata, senza la necessità di riedizione del presente documento ma solamente tramite opportuno avviso. Si può quindi affermare che l'opzionalità di tali componenti è solamente transitoria.

² La descrizione della semantica verrà discussa nel documento REGOLE E PROCEDURE DI NOMENCLATURA. Essa pertanto non verrà ulteriormente discussa nel presente documento.

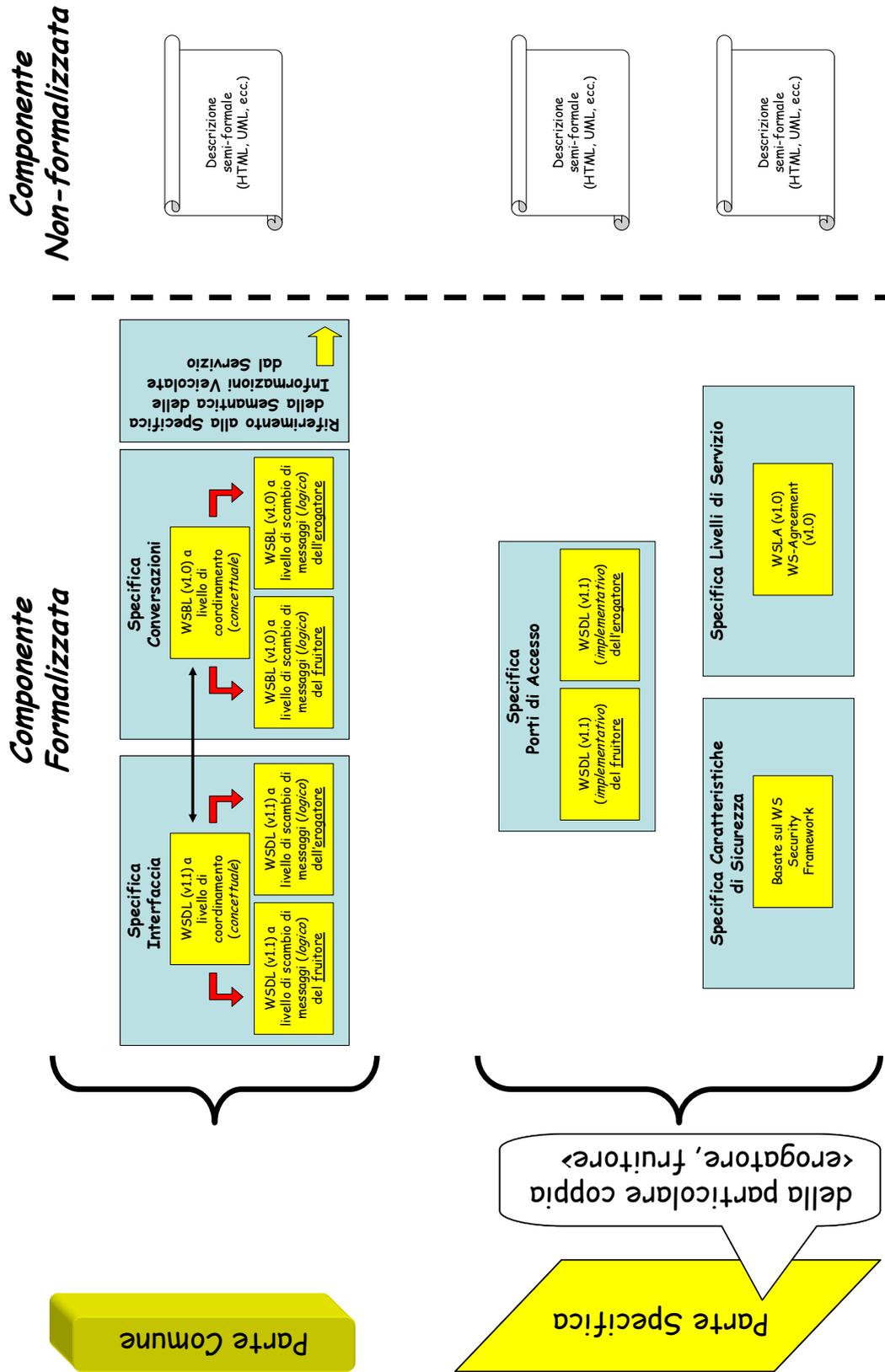


Figura 5. Struttura dell'Accordo di Servizio

Inoltre, la crescita delle tecnologie Web Services quasi certamente vedrà nel prossimo futuro l'evoluzione degli standard disponibili su cui basare la formalizzazione dell'AS; in particolare, è prevedibile che vengano proposti nuovi linguaggi con i quali descrivere la Specifica delle Conversazioni e dei Livelli di Servizio³. Le conseguenti e necessarie evoluzioni delle presenti specifiche, in cui verranno indicati i nuovi linguaggi con cui formalizzare i componenti dell'AS, verranno gestite attraverso la pubblicazione di nuove versioni del presente documento.

³ In particolare è ipotizzabile che in futuro WSBL, che è stato necessario definire appositamente per la Specifica delle Conversazioni, venga sostituito da un linguaggio standard; analogamente WSLA e WS-Agreement, che nella versione attuale delle specifiche sono suggeriti per la Specifica dei Livelli di Servizio, verranno in futuro sostituiti dalle loro naturali evoluzioni.

6. SPECIFICA FUNZIONALE DEL SERVIZIO

La Specifica Funzionale del Servizio è quella parte dell'Accordo di Servizio che descrive l'interfaccia del servizio e le conversazioni supportate⁴. Come introdotto in § 5.2, questa parte è riusabile tra differenti AS (in tutti i casi multi-fruitore e/o multi-erogatore), e pertanto a rigore fa parte della parte comune dell'Accordo di Servizio.

Essa si compone di 6 elementi formalizzati:

- descrizione WSDL a livello di scenario di coordinamento (abbreviato in *WSDL concettuale*⁵);
- descrizione WSDL dell'erogatore a livello di scambio elementare di messaggi (abbreviato in *WSDL logico dell'erogatore*);
- descrizione WSDL del fruitore a livello di scambio elementare di messaggi (abbreviato in *WSDL logico del fruitore*);
- descrizione WSBL a livello di scenario di coordinamento (abbreviato in *WSBL concettuale*); WSBL – Web Service Behavior Language, verrà introdotto in § 6.3.2⁶;
- descrizione WSBL dell'erogatore a livello di scambio elementare di messaggi (abbreviato in *WSBL logico dell'erogatore*);
- descrizione WSBL del fruitore a livello di scambio elementare di messaggi (abbreviato in *WSBL logico del fruitore*).

Accanto a questi documenti è possibile (opzionale) avere altri documenti che descrivono in modo non formalizzato, ma comunque rigoroso, la funzionalità del servizio, commentando e complementando i documenti formali. Come linea guida si raccomanda di produrre:

- un diagramma che descrive il modello informativo sotteso dal servizio [Chessman&Daniels] e che sia annotato con la specifica delle pre- e post-condizioni di ogni operazione, secondo l'approccio "Design by Contract" [Chessman&Daniels];
- un diagramma che visualmente descrive il WSBL concettuale;
- due diagrammi che visualmente descrivono i due WSBL logici dell'erogatore e del fruitore;
- un documento testuale che fornisce ulteriori commenti sul significato delle operazioni offerte dal servizio.

⁴ Come precedentemente descritto, in realtà è compreso anche il Riferimento alla Semantica dell'Informazione Veicolata, ma esso non viene ulteriormente discusso nel presente documento. Si rimanda al documento REGOLE E PROCEDURE DI NOMENCLATURA per gli opportuni dettagli.

⁵ Come precedentemente descritto in § 5.2, i termini "concettuale", "logico" e "fisico" vengono usati in analogia alla progettazione delle basi di dati per indicare differenti modi di utilizzare una sintassi XML a differenti livelli di astrazione (maggiore astrazione a livello concettuale, maggiori dettagli implementativi a livello fisico).

⁶ WSBL si basa su un modello formale basato su macchine a stati finiti; tale modello formale verrà introdotto in § 6.2.1.1 per la parte a livello concettuale ed il § 6.2.1.2 per quella a livello logico; successivamente verrà presentata la sintassi XML che effettivamente rappresenta queste macchine a stati finiti in documenti XML.

Nel corso della presente sezione, verranno mostrati degli esempi della parte formalizzata. In particolare (i) per la specifica delle interfacce verranno mostrati gli estratti XML presentati in Codice 1, Codice 2 e Codice 3, mentre (ii) per la specifica delle conversazioni verranno mostrate Figura 7, Figura 8, Codice 4, Codice 5 e Codice 6.

6.1. Specifica delle Interfacce

L'interfaccia del servizio, intesa come insieme delle operazioni offerte, deve essere descritta in WSDL – Web Service Description Language, versione 1.1. In particolare, tre differenti descrizioni WSDL devono essere presenti:

- Descrizione WSDL a “livello concettuale”, che descrive le operazioni offerte dal servizio in termini di scenari elementari di coordinamento. Questo documento WSDL contiene solo i costrutti `types`, `message`, `operation` e `port type`, e le `operation` possono essere di tutti e 4 i tipi previsti dal linguaggio: *one-way*, *request/reply*, *notification* e *notification/response*, corrispondenti esattamente ai 4 scenari elementari di coordinamento ricordati in § 4.

A titolo di semplice esempio, si supponga di voler descrivere un servizio (del quale per semplicità non ci interessano i parametri dei messaggi ed il tipo di dati veicolati attraverso tali parametri/messaggi) che offre due operazioni (a livello di scenari elementari di coordinamento), una richiesta/risposta A di tipo asincrono ed una notificazione con risposta B; il servizio presenta il WSDL “concettuale” qui riportato:

```
<definitions xmlns=http://schemas.xmlsoap.org/wsdl/
  xmlns:soap=http://schemas.xmlsoap.org/wsdl/soap/
  xmlns:http=http://schemas.xmlsoap.org/wsdl/http/
  xmlns:xs=http://www.w3.org/2001/XMLSchema
  xmlns:soapenc=http://schemas.xmlsoap.org/soap/encoding/
  xmlns:mime=http://schemas.xmlsoap.org/wsdl/mime/
  xmlns:this=http://new.webservice.namespace
  targetNamespace="http://new.webservice.namespace">

  <!-- Non vengono descritti i types per semplicita'.
    Dell'esempio non sono interessanti i tipi ed i nomi
    dei parametri
  -->
  <types/>

  <!-- Messaggi necessari per definire le operazioni.
    Il nome di ogni messaggio indica la sua funzione
    e l'operazione per cui e' definito. Es.:
    richiesta_RichiestaRispostaAsincrona_A e' il messaggio
    inviato come richiesta per l'operazione
    RichiestaRispostaAsincrona_A
  -->
  <message name="richiesta_RichiestaRispostaAsincrona_A"/>
  <message name="risposta_RichiestaRispostaAsincrona_A"/>
  <message name="segnalazione_NotificazioneRisposta_B"/>
  <message name="risposta_NotificazioneRisposta_B"/>

  <!-- Definizione delle operazioni/scenari elementari di
    coordinamento e del servizio
```

```
-->
<portType name="EsempioWSDLConcettuale">
  <operation name="RichiestaRispostaAsincrona_A">
    <input message="this:richiesta_RichiestaRispostaAsincrona_A"/>
    <output message="this:risposta_RichiestaRispostaAsincrona_A"/>
  </operation>
  <operation name="NotificazioneRisposta_B">
    <output message="this:segnalazione_NotificazioneRisposta_B"/>
    <input message="this:risposta_NotificazioneRisposta_B"/>
  </operation>
</portType>
</definitions>
```

Codice 1. Descrizione WSDL concettuale di un semplice servizio

- Documenti WSDL a “livello logico” dell’erogatore e del fruitore, che descrivono le operazioni offerte dai Web Service da dispiegare concretamente sul lato erogatore e sul lato fruitore affinché il servizio sia effettivamente realizzato. In base all’architettura di SPCoop, sono possibili solamente tre tipologie di scambio elementare di messaggi, e quindi ad un servizio a livello concettuale corrispondono (in presenza di operazioni richiesta/risposta asincrone e di notificazioni) due Web Service, uno sul lato erogatore ed uno sul lato fruitore, quest’ultimo necessario per le operazioni di risposta asincrone e le notificazioni. Questi due documenti WSDL contengono solo i costrutti `types`, `message`, `operation` e `port type`, e le `operation` possono essere solamente di 2 dei 4 i tipi previsti dal linguaggio: *one-way* e *request/reply*, che possono essere messe in corrispondenza con i 3 tipi di scambio elementare di messaggi ricordati in § 4. Nel caso del servizio precedentemente descritto a livello concettuale, vengono riportati i WSDL “logici” corrispondenti:

```
<!-- *****
      EROGATORE
      ***** -->
<definitions xmlns=http://schemas.xmlsoap.org/wsdl/
  xmlns:soap=http://schemas.xmlsoap.org/wsdl/soap/
  xmlns:http=http://schemas.xmlsoap.org/wsdl/http/
  xmlns:xs=http://www.w3.org/2001/XMLSchema
  xmlns:soapenc=http://schemas.xmlsoap.org/soap/encoding/
  xmlns:mime=http://schemas.xmlsoap.org/wsdl/mime/
  xmlns:this=http://new.webserviceEROGATORE.namespace
  targetNamespace="http://new.webserviceEROGATORE.namespace">

  <types/>

  <!-- Il Web Service sul lato erogatore offre una operazione
        one-way per ricevere la richiesta dell'operazione
        RichiestaRispostaAsincrona_A e una operazione one-way
        per ricevere la risposta in seguito all'operazione
        NotificazioneRisposta_B
  -->
  <message name="richiesta_RichiestaRispostaAsincrona_A"/>
  <message name="risposta_NotificazioneRisposta_B"/>

  <portType name="Erogatore">
    <operation name="RichiestaRispostaAsincrona_A">
      <input message="this:richiesta_RichiestaRispostaAsincrona_A"/>
    </operation>
    <operation name="NotificazioneRisposta_B">
      <input message="this:risposta_NotificazioneRisposta_B"/>
    </operation>
  </portType>
</definitions>
```

```
</operation>  
</portType>  
</definitions>
```

Codice 2. Descrizione WSDL logica dell'erogatore

```
<!-- *****  
FRUITORE  
***** -->  
<definitions xmlns=http://schemas.xmlsoap.org/wsdl/  
  xmlns:soap=http://schemas.xmlsoap.org/wsdl/soap/  
  xmlns:http=http://schemas.xmlsoap.org/wsdl/http/  
  xmlns:xs=http://www.w3.org/2001/XMLSchema  
  xmlns:soapenc=http://schemas.xmlsoap.org/soap/encoding/  
  xmlns:mime=http://schemas.xmlsoap.org/wsdl/mime/  
  xmlns:this=http://new.webserviceFRUITORE.namespace  
  targetNamespace="http://new.webserviceFRUITORE.namespace">  
  
  <types/>  
  
  <!-- Il Web Service sul lato fruitore offre una operazione  
  one-way per ricevere la risposta asincrona dell'operazione  
  RichiestaRispostaAsincrona_A e una operazione one-way  
  per ricevere la segnalazione dell'operazione  
  NotificazioneRisposta_B  
  -->  
  
  <message name="risposta_RichiestaRispostaAsincrona_A"/>  
  <message name="segnalazione_NotificazioneRisposta_B"/>  
  <portType name="Fruitore">  
    <operation name="RichiestaRispostaAsincrona_A">  
      <input message="this:risposta_RichiestaRispostaAsincrona_A"/>  
    </operation>  
    <operation name="Notificazione_B">  
      <input message="this:segnalazione_NotificazioneRisposta_B"/>  
    </operation>  
  </portType>  
</definitions>
```

Codice 3. Descrizione WSDL logica del fruitore

6.2. Specifica delle Conversazioni

Oltre alla specifica dell'interfaccia, un accordo di servizio deve descrivere anche il cosiddetto *protocollo di business*, cioè la specifica di quali possibili operazioni/sequenze di messaggi sono supportate dal servizio. Tali sequenze di messaggi sono chiamate *conversazioni* [IJCIS2004], e sono estremamente importanti ai fini della comprensione del funzionamento di un servizio, poiché tipicamente le invocazioni delle operazioni di un servizio (e quindi le possibili sequenze di messaggi) rispettano certi vincoli di ordinamento.

Le possibili conversazioni vengono modellate attraverso una macchina a stati finiti (FSM – Finite State Machine) [IJCIS2004,IJCIS2005], in cui gli stati rappresentano i differenti stati

“astratti”⁷ nei quali un servizio può trovarsi durante l’interazione con il suo fruitore, e le transizioni avvengono principalmente (ma non solo) a causa dell’invocazione delle operazioni. La forma esatta della macchina a stati finiti è leggermente differente a seconda se le conversazioni sono descritte a livello concettuale od a livello logico.

La specifica delle conversazioni richiede la definizione di un modello, di una sintassi XML per la descrizione delle conversazioni basate su tale modello, e di regole su come effettivamente utilizzare tale modello e sintassi XML all’interno della parte comune di una AS. Ne seguito, verranno presentate queste tre componenti secondo il seguente dettaglio:

- (a) In § 6.2.1 verrà descritto il modello su cui si basa la specifica delle conversazioni; in particolare in § 6.2.1.1 verrà dettagliato il modello a livello concettuale, ed in § 6.2.1.2 quello a livello logico. Durante la presentazione del modello, a puro scopo chiarificatore, verrà utilizzato un esempio, ispirato da [IJCIS2004], riguardante un dominio generico, non collegato all’ambito dell’e-Government. Tale esempio serve solamente per introdurre i concetti.
- (b) In § 6.2.2 verrà descritto come, sulla base del modello appena introdotto, devono essere specificate le conversazioni in un AS. Qui verrà ripreso lo stesso esempio utilizzato in § 6.1, mostrando la specifica delle conversazioni. In particolare verranno mostrati sia i diagrammi che visualmente descrivono le conversazioni (componente non formalizzata, cfr. Figura 7 e Figura 8) che le descrizioni WSBL (componente formalizzata, cfr. Codice 4, Codice 5 e Codice 6).
- (c) Il dettaglio della sintassi XML appositamente definita per la specifica delle conversazioni, indicata come WSBL – Web Service Behavior Language, ovvero l’XML Schema Definition, verrà invece riportato in § 6.3.2.

Questa stessa struttura è la stessa seguita per la specifica delle interfacce. In tal caso, però, si è omessa la presentazione del modello, in quanto è quello che sottende WSDL e che si assume già ampiamente noto al lettore (cfr. punto (a)); come utilizzare WSDL all’interno dell’AS (cfr. punto (b)) è quello descritto in § 6.1, ed infine considerazioni tecniche su WSDL (punto (c)) sono riportate in § 6.3.1.

6.2.1. Modello per la Specifica delle Conversazioni

6.2.1.1. Livello Concettuale

A livello concettuale, le conversazioni vengono modellate come una macchina a stati finiti; in Figura 6 viene mostrata una specifica concettuale, semplificata ed ipotetica, del servizio offerto da una libreria elettronica; gli stati sono etichettati con un nome significativo, e le transizioni con eventi corrispondenti all’invocazione delle operazioni. Informalmente⁸,

⁷ Si parla di stati “astratti” in quanto essi non rappresentano lo stato effettivo dell’applicazione software che realizza il servizio (non descrivono nessun aspetto di carattere implementativo), ma piuttosto rappresentano punti/stati in cui il servizio si trova durante il “dialogo” (interazione) con la sua controparte.

⁸ Per semplicità espositiva, e data la vasta diffusione delle macchine a stati finiti in molti contesti dell’informatica, si adotta un approccio rigoroso alla descrizione ma non formale e matematico. Il lettore interessato può riferirsi a [IJCIS2004] e [IJCIS2005] per tutti gli aspetti formali.

quando un evento e accade durante una conversazione nello stato src : (i) se src ha una transizione in uscita tr etichettata con e , allora la conversazione si muove nello stato di destinazione di tr , altrimenti (ii) se lo stato src non ha transizioni uscenti etichettate con e , allora la conversazione rimane in src e l'evento viene "consumato" senza effetti. Uno stato è etichettato come iniziale ed alcuni stati sono etichettati come finali, intendendo che in tali stati il dialogo (interazione) con il servizio può essere terminato. In questo modo, tutte le possibili sequenze di transizioni da uno stato iniziale ad uno stato finale sono ricavabili da questo modello compatto, e rappresentano tutte e sole le conversazioni possibili con il servizio.

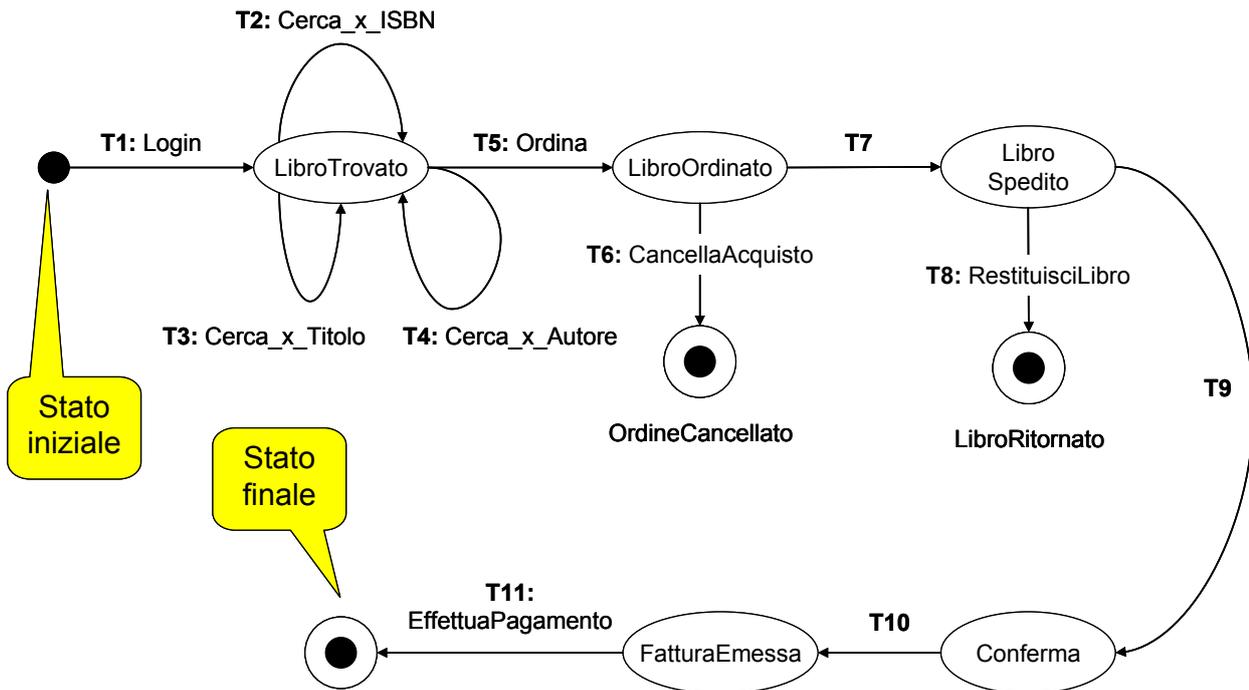


Figura 6. Macchina a stati finiti per la specifica delle conversazioni

Per meglio classificare le transizioni, esse sono etichettate non solamente dall'operazione corrispondente, ma anche da altre possibili proprietà; nel seguito esse vengono descritte in dettaglio.

Proprietà di Attivazione. La maggior parte delle transizioni avviene in seguito all'invocazione di un'operazione. Ma in certi casi, ci sono transizioni che possono avvenire senza l'invocazione di un'operazione, od almeno senza un'esplicita invocazione da parte del fruitore. In Figura 6 questo è il caso della transizione T7 tra gli stati LibroOrdinato e LibroSpedito, che avviene su iniziativa *interna* del servizio. Nel caso di transizioni implicite, una caratteristica importante è la sua temporizzazione, cioè il fatto che la transizione avvenga automaticamente dopo un certo periodo di tempo o quando un determinato istante temporale sia raggiunto.

Nel modello proposto, ogni transizione è etichettata con una *proprietà di attivazione*, che specifica un modo, un evento ed una eventuale condizione temporale di attivazione:

- Modo di attivazione: l'attivazione della transizione è esplicita da parte del fruitore attraverso l'invocazione di un'operazione, ovvero è implicita da parte dell'erogatore.

- Evento: se il modo è esplicito, l'evento è l'invocazione di un'operazione; se il modo è implicito, l'evento è di tipo temporale.

<i>Transizione</i>	<i>Condizione Temporale</i>
T1 ... T5	TRUE
T6	C- invoke (<, end (T5) +7)
T7	M- invoke (≥, end (T5) +7)
T8	C- invoke (<, end (T7) +30)
T9	M- invoke (≥, end (T7) +30)
T10	M- invoke (≥, end (T9) +2)
T11	C- invoke (<, end (T10) +10)

Tabella 3. Condizioni temporali per la specifica delle conversazioni⁹

- Condizione temporale di attivazione: specifica vincoli temporali. Questi vincoli sono espressi come formule di un linguaggio avente i seguenti costrutti:
 - begin(t) e end(t): indicano il timestamp d'inizio e di fine dell'ultima invocazione della transizione t nella presente conversazione;
 - M-invoke: predicato temporale che indica quando una transizione implicita deve essere attivata. M-invoke è usato per specificare eventi temporali;
 - C-invoke: predicato temporale che indica una scadenza e/o finestra temporale entro cui una transizione può essere attivata. C-invoke è usato per le pre-condizioni temporali di una transizione;
 - condizione temporale: ha la forma Pred (boolop, d), dove Pred può essere M-invoke o C-invoke, boolop è un'operatore booleano di confronto (=, <, >, ≤, ≥) e d è un valore temporale assoluto o relativo (ovvero begin(t) o end(t)).
M-invoke (boolop, d) significa che la transizione è automaticamente attivata quando la condizione:

<tempo corrente> boolop d

⁹ I valori numerici assoluti sono in giorni.

è valutata essere vera; si assume che esista la possibilità di conoscere <tempo corrente> (la stessa ipotesi sarà presente parlando dei Service Level Agreement, § 7, ed è ammissibile in quanto il Sistema Pubblico di Connettività fornisce questa funzione a tutti i sistemi che su di lui si appoggiano).

C-invoke (boolop, d) significa che la transizione può essere attivata solo se la condizione:

<tempo corrente> boolop d

è valutata essere vera.

A scopo chiarificatore, in Tabella 2 sono mostrate le condizioni temporali nel caso del servizio di Figura 6; la condizione associata a T6 significa che la transizione può essere invocata solamente entro 7 giorni dal completamento di T5 (ovvero deve essere $begin(T6) < end(T5) + 7$).

Astrazioni di Completamento. Quando si fruisce di un servizio, alcune operazioni possono talvolta essere “cancellate”, nel senso che *dal punto di vista del fruitore*¹⁰ tutti gli effetti precedentemente prodotti possono essere annullati. Questo annullamento tipicamente è disponibile come un’operazione, che però va invocata entro una determinata finestra temporale. Per modellare questi aspetti “transazionali” delle operazioni offerte da un servizio, nel modello proposto è possibile annotare ogni transizione con uno dei seguenti tipi:

- Senza Effetti (*effect-less*): denota una transizione che non ha effetti permanenti dal punto di vista del fruitore. Cancellare la transizione non richiede l’esecuzione di nessuna operazione particolare.
- Compensabile (*compensatable*): denota una transizione che ha degli effetti che possono essere esplicitamente cancellati invocando un’operazione di compensazione.
- Definitiva (*definite*): denota una transizione i cui effetti sono permanenti.

Come esempio, si riporta in Tabella 4 le proprietà di completamento per le transizioni nel caso del servizio della libreria elettronica.

<i>Transizio ne</i>	<i>Proprietà</i>	<i>Compensa zione</i>
T1	Definitiv a	
T2, T3, T4	Senza Effetti	

¹⁰ E’ importante sottolineare che il significato della “cancellazione degli effetti” di un’operazione di un servizio non è quello del rollback di una transazione di un database. Dal punto di vista dell’erogatore, la cancellazione degli effetti può infatti essere molto costosa, richiedere risorse specializzate e spesso richiede nuove operazioni interne. E’ dal punto di vista dei fruitore che l’operazione appare “cancellata”. Ad esempio si consideri la cancellazione di una prenotazione: mentre dal punto di vista del fruitore questo equivale all’annullamento dell’operazione di prenotazione, l’erogatore di fatto non fa il semplice rollback della transazione di prenotazione, ma avvia una nuova transazione che compensa gli effetti della prenotazione (rendendo disponibile nuovamente la risorsa riservata); nell’erogatore rimane cioè traccia sia dell’operazione che della compensazione necessaria al suo annullamento.

T5	Compensabile	T6
T6	Definitiva	
T7	Compensabile	T8
T8, T9, T10, T11	Definitiva	

Tabella 4. Proprietà di completamento per la specifica delle conversazioni

6.2.1.2. Livello Logico

Anche a livello logico, le conversazioni del Web Service dell'erogatore e del fruitore vengono modellate come macchine a stati finiti; in questo caso però le transizioni della FSM sono etichettate solamente con eventi di emissione/ricezione dei messaggi. Tali transizioni non sono etichettate con nessuna proprietà, ma solamente con i messaggi scambiati; derivando però tali messaggi dalle operazioni definite a livello concettuale (ed essendo di fatto tali messaggi "incapsulati" all'interno delle operazioni definite nei Web Service secondo i 3 possibili schemi di scambio elementare), è immediato risalire a quali proprietà essi inducono.

Più specificatamente, le FSM a livello logico possono non presentare nomi sugli stati, e le transizioni sono etichettate con i nomi dei messaggi, seguiti da un simbolo che denota se il messaggio viene ricevuto o viene spedito; tali simboli sono: >> per indicare l'evento di ricezione, e << per indicare l'evento di emissione; entrambi i simboli vanno prefissi al nome della transizione/messaggio che la etichetta.

A livello grafico, alcuni stati saranno rappresentati come un quadrato, ad indicare che essi sono introdotti per la prima volta al livello logico per rappresentare le possibili sequenze di scambio di messaggi ammissibili, mentre altri saranno rappresentati con lo stesso simbolo usato per gli stati a livello concettuale, proprio per enfatizzare che sono stati del Web Service (dell'erogatore o del fruitore) che hanno un significato non solo in termini di scambio di messaggi, ma anche a livello più alto.

6.2.2. Specifica delle Conversazioni nell'AS

Come deriva dal modello appena presentato, in un AS la specifica delle conversazioni viene espressa a due livelli: concettuale e logico, quest'ultimo distinguendo il lato erogatore ed il lato fruitore.

A livello più astratto di descrizione del servizio, accanto al WSDL concettuale, deve essere presente una FSM che descrive il protocollo del servizio, secondo il modello introdotto in § 6.2.1.1 Questa descrizione concettuale del protocollo del servizio viene espressa (i) nella parte formalizzata attraverso un documento WSBL, e (ii) nell'altra attraverso un diagramma

simile a quello di Figura 6, eventualmente corredato da opportune tabelle per le proprietà della transizioni (simili a Tabella 3 e Tabella 4).

A livello logico, accanto ai due WSDL dell'erogatore e del fruitore, devono essere presenti le rispettive FSM, secondo il modello introdotto in § 6.2.1.2. Queste vengono espresse (i) nella parte obbligatoria attraverso due documenti WSBL, e (ii) nella parte opzionale attraverso due diagrammi.

Si consideri nuovamente l'esempio introduttivo presentato in § 6.1, e si immagini che il protocollo di tale servizio preveda prima l'invocazione dell'operazione `RichiestaRispostaAsincrona_A`, e solo successivamente la `NotificazioneRisposta_B`. Si supponga inoltre che la prima operazione sia definitiva, e che la seconda debba essere eseguita entro dieci giorni dalla precedente. E' immediato osservare come queste informazioni, che dipendono dalla logica applicativa del servizio, e che sono essenziali affinché il servizio venga correttamente fruito, non sono esprimibili nei documenti WSDL. Vedremo ora come attraverso il modello per la specifica delle conversazioni, sia possibile esprimerle, attraverso documenti WSBL (parte formalizzata) ed attraverso diagrammi visuali (parte non formalizzata).

La specifica WSBL concettuale del servizio, assumendo che il suo protocollo sia quello appena descritto, è la seguente:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsbl:ConceptualBehavior xmlns:wsbl="http://cnipa.it/wsbl"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="...">
  <name>ESEMPIO WSBL CONCETTUALE</name>
  <states>
    <state-initial name="start"/>
    <state-final name="B_Fatto"/>
    <state name="A_Fatto"/>
  </states>
  <transitions>
    <transition source="start" target="A_Fatto" name="T1">
      <activationMode>EXPLICIT</activationMode>
      <completionMode>
        <mode>DEFINITIVE</mode>
      </completionMode>
      <events>
        <event name="RichiestaRispostaAsincrona_A"/>
      </events>
    </transition>
    <transition source="A_Fatto" target="B_Fatto" name="T2">
      <activationMode>IMPLICIT</activationMode>
      <completionMode>
        <mode>EFFECTLESS</mode>
      </completionMode>
      <events>
        <event name="NotificazioneRisposta_B"/>
      </events>
      <temporalCondition>
        <predicate>M-INVOKE</predicate>
        <boolop>GEQ</boolop>
        <data>end(T1)+10</data>
      </temporalCondition>
    </transition>
  </transitions>
</wsbl:ConceptualBehavior>
```

Codice 4. Descrizione WSBL concettuale delle conversazioni

A livello visuale, in Figura 7 è riportata la FSM, con le transizioni etichettate con le relative proprietà. Si rimarca che questa specifica complementa la specifica concettuale dell'interfaccia espressa con il documento WSDL di Codice 1.

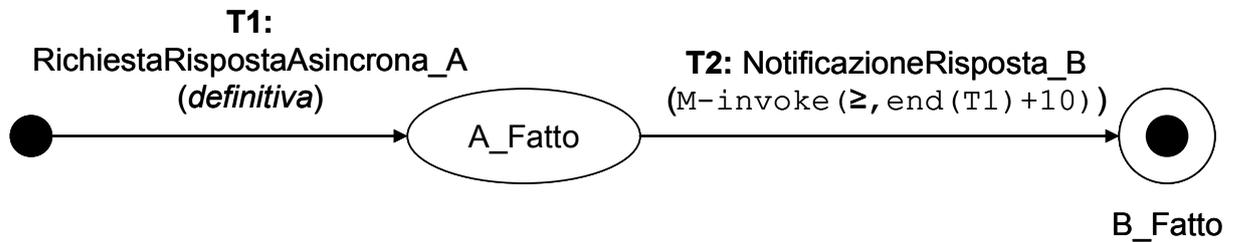


Figura 7. Specifica concettuale delle conversazioni

A livello logico, i due documenti WSBL dell'erogatore e del fruitore sono i seguenti:

```

<?xml version="1.0" encoding="UTF-8"?>
<wsbl:MessageBehavior xmlns:wsbl=http://cnipa.it/wsbl
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="...">
  <name>EROGATORE</name>
  <states>
    <state-initial name="start"/>
    <state-final name="B_Fatto"/>
    <state name="1"/>
    <state name="A_Fatto"/>
    <state name="3"/>
  </states>
  <messages>
    <message source="start" target="1"
name="richiesta_RichiestaRispostaAsincrona_A" type="RECEIVE"/>
    <message source="1" target="A_Fatto"
name="risposta_RichiestaRispostaAsincrona_A" type="SEND"/>
    <message source="A_Fatto" target="3"
name="segnalazione_NotificazioneRisposta_B" type="SEND"/>
    <message source="3" target="B_Fatto"
name="risposta_NotificazioneRisposta_B" type="RECEIVE"/>
  </messages>
</wsbl:MessageBehavior>
  
```

Codice 5. Descrizione WSBL logica dell'erogatore

```

<?xml version="1.0" encoding="UTF-8"?>
<wsbl:MessageBehavior xmlns:wsbl=http://cnipa.it/wsbl
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="...">
  <name>FRUITORE</name>
  <states>
    <state-initial name="start"/>
    <state-final name="B_Fatto"/>
    <state name="1"/>
    <state name="A_Fatto"/>
    <state name="3"/>
  </states>
  <messages>
    <message source="start" target="1"
name="richiesta_RichiestaRispostaAsincrona_A" type="SEND"/>
    <message source="1" target="A_Fatto"
name="risposta_RichiestaRispostaAsincrona_A" type="RECEIVE"/>
    <message source="A_Fatto" target="3"
name="segnalazione_NotificazioneRisposta_B" type="RECEIVE"/>
  </messages>
  
```

```
<message source="3" target="B_Fatto"
  name="risposta_NotificazioneRisposta_B" type="SEND"/>
</messages>
</wsbl:MessageBehavior>
```

Codice 6. Descrizione WSBL logica del fruitore

Le rappresentazioni visuali delle due FSM sono mostrate in Figura 8; esse complementano i due documenti WSDL presentati in Codice 2 e Codice 3. E' possibile riconoscere gli stati introdotti per correttamente rappresentare lo scambio dei messaggi, e quelli che corrispondono agli stati del livello concettuale.

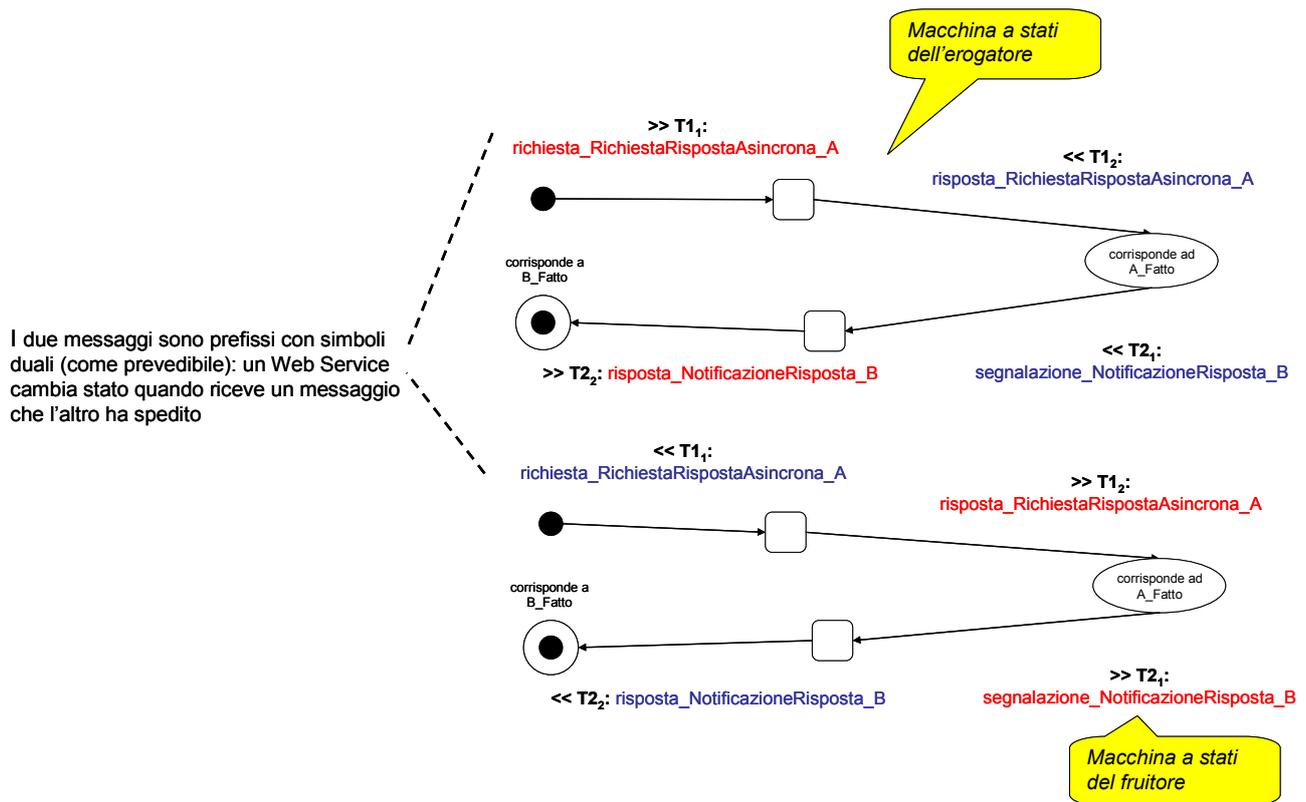


Figura 8. Specifica logica delle conversazioni dell'erogatore e del fruitore. Per immediatezza visiva, l'evento di ricezione ha il messaggio in rosso e quello di emissione in blu (oltre ai simboli prescritti, rispettivamente >> e <<)

6.3. Rappresentazione in XML

In questa sezione vengono forniti dettagli tecnici sulla rappresentazione XML della Specifica delle Interfacce e delle Conversazioni

6.3.1. Raccomandazioni per l'utilizzo di WSDL

Come precedentemente descritto in § 6.1, la Specifica delle Interfacce prevede l'utilizzo di WSDL (Web Service Description Language, versione 1.1), attraverso tre documenti distinti, uno a livello concettuale e due a livello logico

Ai fini di migliorare la modularizzazione, è raccomandabile che la parte di definizione dei types e dei message, che è in comune tra tutti e tre i documenti, fosse presentata in un documento a parte, importato da tutti e 3 i WSDL. Pertanto è consigliabile che effettivamente i documenti WSDL che descrivono l'interfaccia del servizio siano 4: (i) WSDL definitorio, (ii) WSDL "concettuale", (iii) WSDL "logico" dell'erogatore e (iv) WSDL "logico" del fruitore. Anche nel caso di servizi che non richiedono Web Service sul lato fruitore (servizi sincroni ovvero senza operazioni di notificazione), quest'ultimo WSDL deve essere presente come documento vuoto, ad indicare appunto che sul lato fruitore concretamente non è necessario realizzare nulla di complesso¹¹ per usufruire del servizio. In Figura 9 vengono riportati i 4 documenti WSDL nel semplice esempio precedente utilizzato e le loro relazioni.

Per quanto riguarda le convenzioni sintattiche da utilizzare, si raccomandano le seguenti regole; si assume che il servizio offre un'operazione di nome <NomeOperazione>, in cui è buona norma che <NomeOperazione> sia significativo e fornisca un'intuizione del comportamento dell'operazione. Allora:

1. Per ogni messaggio dell'operazione offerta dal servizio, nel documento Types.xsd (contenente le definizioni da importare nei tre file WSDL) vi deve essere la definizione di un complexType che definisce la struttura dei parametri, eventualmente con la definizione di eventuali tipi necessari per definire i parametri (si confronti in Figura 9 la definizione

```
<xs:complexType
name="richiesta_RichiestaRispostaAsincrona_A_Type"/>
```
2. Il nome di questo tipo avrà la forma <Direzione>_<TipologiaScenario>_<NomeOperazione>_Type con:
 - o <Direzione> può assumere uno dei 3 valori: richiesta, risposta, segnalazione;
 - o <TipologiaScenario> può assumere uno dei 5 valori: RichiestaSenzaRisposta, RichiestaRispostaSincrona, RichiestaRispostaAsincrona, NotificazioneRisposta, NotificazioneSenzaRisposta.
3. Sempre per ogni messaggio dell'operazione, in Types.xsd vi deve essere la definizione di un element, di tipo il complexType di cui ai punti 1 e 2, il cui nome avrà la forma <Direzione>_<TipologiaScenario>_<NomeOperazione>.

Ad esempio, nel caso che l'operazione sia RestituisciIndirizzo, di tipo richiesta/risposta sincrona, che prende in input nome, cognome e codice fiscale di un

¹¹ Ovviamente il lato fruitore deve realizzare un SOAP client in grado di invocare le operazioni offerte dal Web Service dell'erogatore. L'affermazione "non deve realizzare nulla di complesso" va intesa nel senso che niente di complesso ("logica servente") deve essere realizzato sul lato fruitore oltre a questo programma SOAP client.

cittadino e restituisce un indirizzo¹², allora le definizioni necessarie sono:

```
<xs:complexType
  name="richiesta_RichiestaRispostaSincrona_RestituisciIndirizzo_Type">
  <xs:sequence>
    <xs:element name="Nome" type="xs:string"/>
    <xs:element name="Cognome" type="xs:string"/>
    <xs:element name="CodiceFiscale">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="16"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:element name="richiesta_RichiestaRispostaSincrona_RestituisciIndirizzo"
  type="tns:richiesta_RichiestaRispostaSincrona_RestituisciIndirizzo_Type"/>

<xs:complexType
  name="risposta_RichiestaRispostaSincrona_RestituisciIndirizzo_Type">
  <xs:sequence>
    <xs:element name="Indirizzo" type="FormatoInterscambioType"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="risposta_RichiestaRispostaSincrona_RestituisciIndirizzo"
  type="tns:risposta_RichiestaRispostaSincrona_RestituisciIndirizzo_Type"/>
```

4. Sempre per ogni messaggio dell'operazione offerta dal servizio, nei tre file WSDL saranno presenti le definizioni dei messaggi, che utilizzano come struttura gli element così come definiti al punto 3; il nome di tali messaggi è della forma <Direzione>_<TipologiaScenario>_<NomeOperazione>_Msg (con significato analogo al punto 2).
5. Infine le definizioni del servizio e dei due Web Service dell'erogatore e del fruitore definiscono l'operazione utilizzando i messaggi definiti al punto 4; il nome dell'operazione è della forma <TipologiaScenario>_<NomeOperazione> (con significato analogo al punto 2).

Ad esempio, nel caso del servizio che offre l'operazione RestituisciIndirizzo, nei tre file WSDL saranno presenti le seguenti definizioni:

```
<wsdl:message name="richiesta_RichiestaRispostaSincrona_RestituisciIndirizzo_Msg">
  <wsdl:part name="parameters"
    element="types:richiesta_RichiestaRispostaSincrona_RestituisciIndirizzo"/>
</wsdl:message>
<wsdl:message name="risposta_RichiestaRispostaSincrona_RestituisciIndirizzo_Msg">
  <wsdl:part name="parameters"
    element="types:risposta_RichiestaRispostaSincrona_RestituisciIndirizzo"/>
</wsdl:message>
<wsdl:portType name="Servizio">
  <wsdl:operation name="RichiestaRispostaSincrona_RestituisciIndirizzo">
    <wsdl:input message="tns:richiesta_RichiestaRispostaSincrona_RestituisciIndirizzo_Msg"/>
    <wsdl:output message="tns:risposta_RichiestaRispostaSincrona_RestituisciIndirizzo_Msg"/>
  </wsdl:operation>
</wsdl:portType>
```

¹² Si assume di utilizzare il formato dell'indirizzo che è stato recentemente proposto nell'ambito del progetto congiunto AIPA (ora CNIPA) ed ISTAT [QualitàDati]; tale formato è definito di tipo FormatoInterscambioType.

6.3.2. WSBL – Web Service Behavior Language

Viene riportato l'XML Schema della sintassi che permette di descrivere le conversazioni supportate da un servizio, sia a livello concettuale che logico.

```
<!-- *****
Copyright (C) 2005

Centro Nazionale per l'Informatica nella Pubblica Amministrazione
(CNIPA) - Italy

This XSD is free; you can redistribute it and/or modify it
under the terms of the GNU General Public License as published
by the Free Software Foundation; either version 2, or (at your option)
any later version.

It is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY.
See the GNU Library General Public License for more details.

***** -->
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd=http://www.w3.org/2001/XMLSchema
targetNamespace=http://cnipa.it/wsbl/
xmlns:wsbl="http://cnipa.it/wsbl/">

  <xsd:element name="ConceptualBehavior"
type="wsbl:ConceptualBehaviorType" nillable="false">

    <!-- ##### -->
    <!-- Vincoli univocita' dei nomi -->
    <!-- ##### -->
    <xsd:unique name="uniqueState">
      <xsd:selector xpath="./states/state"/>
      <xsd:field xpath="@name"/>
    </xsd:unique>

    <xsd:unique name="uniqueTransition">
      <xsd:selector xpath="./transitions/transition"/>
      <xsd:field xpath="@name"/>
    </xsd:unique>

    <!-- ##### -->
    <!-- Vincoli integrita' referenziale -->
    <!-- tra STATI e TRANSIZIONI -->
    <!-- ##### -->
    <xsd:key name="keyStateName">
      <xsd:selector xpath="./states/*"></xsd:selector>
      <xsd:field xpath="@name"></xsd:field>
    </xsd:key>
    <xsd:keyref name="checkTransitionSource" refer="wsbl:keyStateName">
      <xsd:selector xpath="./transitions/transition"></xsd:selector>
      <xsd:field xpath="@source"></xsd:field>
    </xsd:keyref>
    <xsd:keyref name="checkTransitionTarget" refer="wsbl:keyStateName">
      <xsd:selector xpath="./transitions/transition"></xsd:selector>
      <xsd:field xpath="@target"></xsd:field>
    </xsd:keyref>

  </xsd:element>

  <xsd:element name="MessageBehavior"
type="wsbl:MessageBehaviorType" nillable="false">
```

```

<!-- ##### -->
<!-- Vincoli univocita' nomi -->
<!-- ##### -->
<xsd:unique name="uniqueState2">
  <xsd:selector xpath="./states/state"/>
  <xsd:field xpath="@name"/>
</xsd:unique>

<xsd:unique name="uniqueMessage">
  <xsd:selector xpath="./messages/message"/>
  <xsd:field xpath="@name"/>
</xsd:unique>

<!-- ##### -->
<!-- Vincoli integrita' referenziale -->
<!-- tra STATI e TRANSIZIONI -->
<!-- ##### -->
<xsd:key name="keyStateName2">
  <xsd:selector xpath="./states/*"></xsd:selector>
  <xsd:field xpath="@name"></xsd:field>
</xsd:key>
<xsd:keyref name="checkMessageSource" refer="wsbl:keyStateName2">
  <xsd:selector xpath="./messages/message"></xsd:selector>
  <xsd:field xpath="@source"></xsd:field>
</xsd:keyref>
<xsd:keyref name="checkMessageTarget" refer="wsbl:keyStateName2">
  <xsd:selector xpath="./messages/message"></xsd:selector>
  <xsd:field xpath="@target"></xsd:field>
</xsd:keyref>

</xsd:element>

<!-- ##### -->
<!-- Definizione tipi -->
<!-- ##### -->

<xsd:complexType name="statesType">
  <xsd:sequence>
    <xsd:element name="state-initial" type="wsbl:StateTypeInitial"
      nillable="false" minOccurs="1" maxOccurs="1"></xsd:element>
    <xsd:element name="state-final" type="wsbl:StateTypeFinal"
      nillable="false" minOccurs="1" maxOccurs="1"></xsd:element>
    <xsd:element name="state" type="wsbl:StateTypeNormal"
      nillable="false" minOccurs="1" maxOccurs="unbounded"></xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="transitionsType">
  <xsd:sequence>
    <xsd:element name="transition" type="wsbl:transitionType"
      minOccurs="2" maxOccurs="unbounded" nillable="false"></xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="messagesTypes">
  <xsd:sequence>
    <xsd:element name="message" minOccurs="1"
      maxOccurs="unbounded" nillable="false">
      <xsd:complexType>
        <xsd:attribute name="name"
          type="wsbl:stringNotEmpty" use="required">
        </xsd:attribute>
        <xsd:attribute name="type"
          type="wsbl:messageOrientationType" use="required">
        </xsd:attribute>
        <xsd:attribute name="source"
          type="wsbl:stringNotEmpty" use="required">
        </xsd:attribute>
        <xsd:attribute name="target"
          type="wsbl:stringNotEmpty" use="required">
        </xsd:attribute>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ConceptualBehaviorType">
    <xsd:sequence>
        <xsd:element name="name" type="wsbl:stringNotEmpty"
            minOccurs="1" maxOccurs="1" nillable="false">
        </xsd:element>
        <xsd:element name="states" type="wsbl:statesType"
            minOccurs="1" maxOccurs="1" nillable="false">
        </xsd:element>
        <xsd:element name="transitions" type="wsbl:transitionsType"
            minOccurs="1" maxOccurs="1" nillable="false">
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MessageBehaviorType">
    <xsd:sequence>
        <xsd:element name="name" type="wsbl:stringNotEmpty"
            minOccurs="1" maxOccurs="1" nillable="false">
        </xsd:element>
        <xsd:element name="states" type="wsbl:statesType"
            minOccurs="1" maxOccurs="1" nillable="false">
        </xsd:element>
        <xsd:element name="messages" type="wsbl:messagesTypes"
            minOccurs="1" maxOccurs="1" nillable="false">
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="completionModeType">
    <xsd:sequence>
        <xsd:element name="mode">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="EFFECTLESS">
                    </xsd:enumeration>
                    <xsd:enumeration value="DEFINITIVE">
                    </xsd:enumeration>
                    <xsd:enumeration value="COMPENSATABLE">
                    </xsd:enumeration>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="compensateMessage" minOccurs="0" maxOccurs="1" >
            <xsd:complexType>
                <xsd:attribute name="name"
                    type="wsbl:stringNotEmpty" use="required">
                </xsd:attribute>
                <xsd:attribute name="type"
                    type="wsbl:messageOrientationType" use="required">
                </xsd:attribute>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="activationType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="IMPLICIT"></xsd:enumeration>
        <xsd:enumeration value="EXPLICIT"></xsd:enumeration>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="messageOrientationType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="RECEIVE"></xsd:enumeration>
        <xsd:enumeration value="SEND"></xsd:enumeration>
    </xsd:restriction>
</xsd:simpleType>

```

```
<xsd:complexType name="eventType">
  <xsd:sequence>
    <xsd:element name="message" minOccurs="0" maxOccurs="1">
      <xsd:complexType>
        <xsd:attribute name="name"
          type="wsbl:stringNotEmpty" use="required">
        </xsd:attribute>
        <xsd:attribute name="type"
          type="wsbl:messageOrientationType" use="required">
        </xsd:attribute>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="name" type="wsbl:stringNotEmpty" use="required">
</xsd:attribute>
</xsd:complexType>

<xsd:complexType name="eventListType">
  <xsd:sequence>
    <xsd:element name="event" type="wsbl:eventType"
      minOccurs="0" maxOccurs="unbounded">
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="guardType">
  <xsd:sequence>
    <xsd:element name="description" type="xsd:string"></xsd:element>
    <xsd:element name="rule" type="wsbl:stringNotEmpty"
      minOccurs="1" maxOccurs="1">
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="name" type="wsbl:stringNotEmpty" use="required">
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="temporalConditionType">
  <xsd:sequence>
    <xsd:element name="predicate" minOccurs="1" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="C-INVOKE"/>
          <xsd:enumeration value="M-INVOKE"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="boolop" minOccurs="1" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="EQ"/>
          <xsd:enumeration value="GR"/>
          <xsd:enumeration value="GEQ"/>
          <xsd:enumeration value="LS"/>
          <xsd:enumeration value="LEQ"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="data" type="wsbl:stringNotEmpty"
      minOccurs="1" maxOccurs="1"/>
    <xsd:element name="description" type="xsd:string"
      minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="transitionType">
  <xsd:sequence>
    <xsd:element name="activationMode" type="wsbl:activationType"
      minOccurs="1" maxOccurs="1">
    </xsd:element>
    <xsd:element name="completionMode" type="wsbl:completionModeType"
      minOccurs="1" maxOccurs="1">
  </xsd:sequence>
</xsd:complexType>
```

```
</xsd:element>
<xsd:element name="guard" type="wsbl:guardType"
  minOccurs="0" maxOccurs="1">
</xsd:element>
<xsd:element name="events" type="wsbl:eventListType" nillable="true"
  minOccurs="0" maxOccurs="1">
</xsd:element>
<xsd:element name="temporalCondition"
  type="wsbl:temporalConditionType"
  minOccurs="0" maxOccurs="1">
</xsd:element>
</xsd:sequence>
<xsd:attribute name="name" type="wsbl:stringNotEmpty" use="required"/>
<xsd:attribute name="source" type="wsbl:stringNotEmpty" use="required"/>
<xsd:attribute name="target" type="wsbl:stringNotEmpty" use="required"/>
</xsd:complexType>

<xsd:complexType name="StateTypeBase" abstract="true">
  <xsd:attribute name="name" type="wsbl:stringNotEmpty" use="required"/>
</xsd:complexType>

<xsd:complexType name="StateTypeInitial" abstract="false">
  <xsd:complexContent>
    <xsd:extension base="wsbl:StateTypeBase">
</xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="StateTypeFinal" abstract="false">
  <xsd:complexContent>
    <xsd:extension base="wsbl:StateTypeBase">
</xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="StateTypeNormal" abstract="false">
  <xsd:complexContent>
    <xsd:extension base="wsbl:StateTypeBase">
</xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:simpleType name="stringNotEmpty">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/></xsd:minLength>
  </xsd:restriction>
</xsd:simpleType>

</xsd:schema>
```

Codice 7. XML Schema di Web Service Behavior Language, per la descrizione delle conversazioni a livello concettuale e logico

7. SPECIFICA DEI LIVELLI DI SERVIZIO

Come precedentemente osservato, nell'AS, a complemento della parte di specifica funzionale del servizio, è presente una componente di specifica degli Accordi sui Livelli di Servizio (SLA – Service Level Agreement), che definisce i livelli/caratteristiche di prestazioni concordate tra erogatore e fruitore, ed il modo di valutarli e misurarli.

Mentre la specifica funzionale è alla base della progettazione e realizzazione del servizio e delle applicazioni che ne fruiscono (e quindi è utilizzata prevalentemente a design-time), la specifica degli SLA è alla base della valutazione, durante l'esercizio e la gestione, dell'aderenza ai livelli garantiti (e quindi è utilizzata prevalentemente a run-time).

La specifica degli SLA si compone di un documento, che può essere non-formalizzato ovvero espresso in uno dei linguaggi XML attualmente proposti allo scopo, quali WSLA (Web Service Level Agreement, v. 1.0) o WS-Agreement (v. 1.1). Tale documento (indipendentemente dall'essere formalizzato o meno) deve definire sia le componenti necessarie a definire gli SLA, che gli SLA veri e propri per una coppia <erogatore, fruitore>; esso quindi appartiene alla parte specifica dell'AS.

Si rimarca che la versione attuale dell'AS prescrive come opzionale la specifica dei livelli di servizio, ed in caso questa venga fornita, non prescrive nessun linguaggio in particolare, né tantomeno la sua formalizzazione. Nel seguito verranno forniti, a puro titolo esemplificativo, alcuni esempi di specifiche formalizzate attraverso una delle proposte attualmente presenti nel mondo della ricerca.

7.1. Concetti di Base

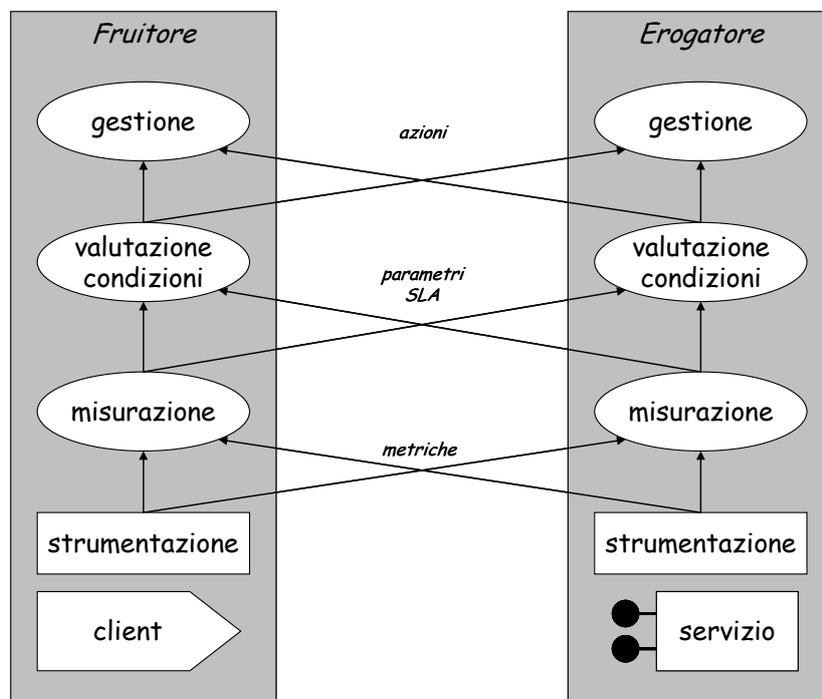
Per poter introdurre gli elementi necessari alla specifica degli SLA, è necessario preliminarmente definire alcuni concetti di base:

- Le *metriche di risorsa* vengono recuperate direttamente dalle risorse gestite dispiegate presso l'erogatore del servizio; esempi di metriche di risorsa sono le variabili MIB dell'IETF Structure of Management Information.
- *Metriche composte* vengono create combinando metriche di risorsa, od altre metriche composte, secondo specifici algoritmi; questo può essere fatto (i) all'interno del dominio dell'erogatore, o (ii) da una terza parte che realizza un servizio di misurazione. Nel contesto di riferimento, si assume che tali metriche composte siano rese accessibili (dall'erogatore e/o dal servizio di misura terzo) attraverso opportune interfacce applicative (ad es., attraverso SOAP).
- I *parametri SLA* contestualizzano le metriche disponibili allo specifico fruitore. Tali parametri hanno spesso associati dei range di variabilità, sulla base dei quali viene valutata la corrispondenza agli *obiettivi sui livelli di servizio* (SLO – Service Level

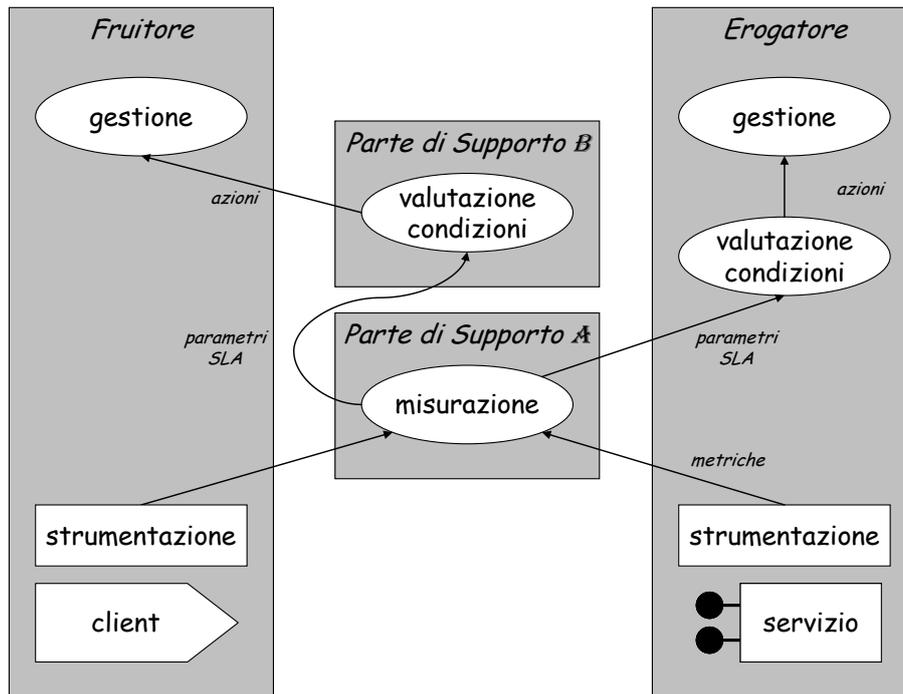
Objective), che definiscono i “desiderata” del fruitore nei confronti dell’erogatore del servizio.

- In Figura 10 è mostrata l’architettura di riferimento per i livelli di servizio. Nel caso base, l’erogatore ed il fruitore, ognuno dotato di specifica strumentazione e di specifici componenti per la misura e la gestione, in modo peer-to-peer gestiscono gli accordi sui livelli di servizio:
 - La *componente di misurazione* riceve le/accede alle metriche dalla/attraverso la strumentazione; le direttive su come misurare e calcolare i particolari valori sono parte della specifica contenuta nell’AS.
 - L’insieme delle metriche che sono utilizzate nella definizione degli accordi di servizio è reso disponibile dalla componente di misurazione come parametri SLA.
 - Una *funzione di valutazione delle condizioni* verifica la corrispondenza dei valori recuperati/calcolati rispetto agli obiettivi concordati.
 - Una *componente di gestione*, attraverso opportune *azioni* correttive, reagisce ad eventuali anomalie.

Nel caso con coinvolgimento con terze parti, definite *parti di supporto*, alcune delle componenti suddette possono essere affidate a soggetti esterni. In questo caso sono ovviamente possibili molteplici configurazioni; si consiglia comunque di mantenere internamente all’erogatore ed al fruitore la componente di gestione, in quanto l’esatta modalità di gestione delle anomalie è una caratteristica propria e caratterizzante dei due soggetti che stipulano l’AS.



(b) Caso base



(a) Coinvolgimento di terze parti

Figura 10. Architettura di Riferimento per i Livelli di Servizio

Mentre allo stato attuale il modello di cooperazione di base sull'architettura di riferimento (a), future evoluzioni prevedono che il SICA offra, attraverso opportuni servizi infrastrutturali, il supporto per l'architettura (b), in cui la terza parte con compiti di misurazione e valutazione delle condizioni sia appunto uno specifico elemento del SICA.

A titolo puramente esemplificativo, viene considerato WSLA, una tecnologia proposta in contesto ricerca ma non standard de-iure¹³, che permette la definizione di tutti gli elementi precedentemente introdotti. In particolare, in Figura 11(a) vengono mostrati, attraverso un diagramma delle classi UML, i principali concetti di WSLA, ad ognuno dei quali corrisponde un equivalente *tag* XML (costruito del linguaggio).

Uno SLA (Service Level Agreement) descritto in WSLA si compone di tre parti:

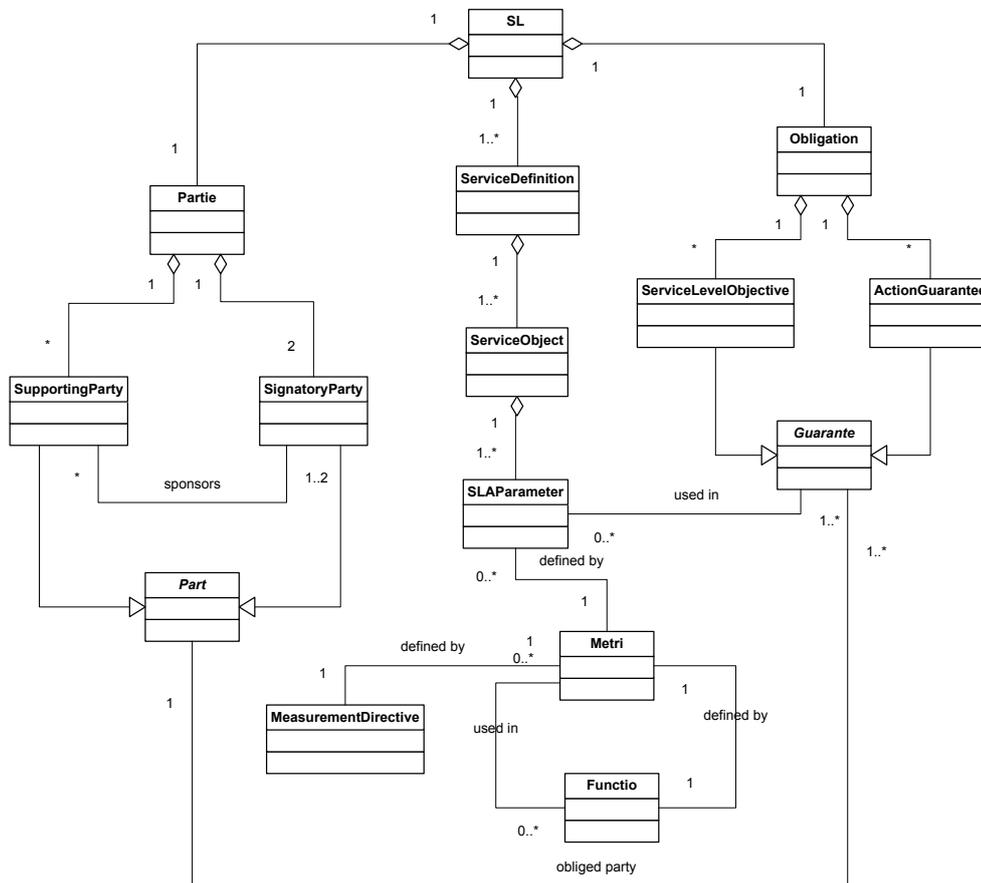
- Definizione delle parti coinvolte (*Parties*), che devono essere le due firmatarie dell'accordo (*SignatoryParty*), cioè l'erogatore ed il fruitore, ed eventuali soggetti di supporto a cui delegare alcune delle funzionalità di misura e monitoraggio (*SupportingParty*, confrontare Figura 10).
- Definizione del servizio (*ServiceDefinition*) di cui si stanno definendo i livelli, che si compone di più oggetti (*ServiceObject*), ognuno dei quali corrisponde ad una delle operazioni offerte dallo stesso. Per ognuno degli oggetti, vengono definiti più parametri (*SLAParameter*), ognuno definito attraverso una metrica (*Metric*). Una

¹³ Attualmente WSLA è in corso di evoluzione in WS-Agreement, nel contesto di iniziative di ricerca sulle tecnologie Grid. WSLA viene qui presentato non come linguaggio prescelto per la specifica degli SLA, ma come esempio di come sia possibile formalizzare la specifica degli SLA attraverso XML.

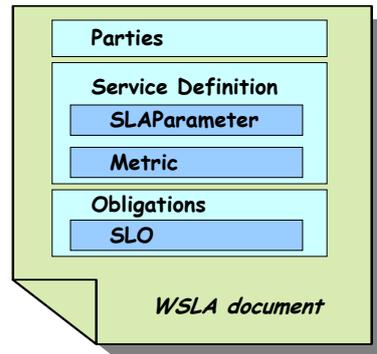
metrica può essere la specifica di come un valore deve essere misurato, attraverso una specifica direttiva (*MeasurementDirective*), o di come può essere computato attraverso una funzione (*Function*) che opera su altre metriche. Le prime sono le metriche di risorsa, mentre le seconde sono le metriche composte.

- Definizione delle obbligazioni (*Obligations*), che possono essere di due tipi:
 - SLO (*ServiceLevelObjective*), che indicano la garanzia sullo stato di un particolare parametro in un dato periodo temporale (ad es., “il tempo medio di risposta è 2 secondi tra le 9.00 e le 18.00”);
 - Azioni (*ActionGuarantee*), che rappresentano la promessa di intraprendere una determinata azione al verificarsi di una determinata situazione.

Sulla base di questi costrutti, la struttura di un eventuale documento WSLA, da introdurre nella parte specifica di un AS, è quella rappresentata in Figura 11(b).



(a) Costrutti e loro relazioni



(b) Struttura di un documento

Figura 11. Principali concetti di WSLA e struttura di un documento (vengono riportati in inglese per rispettare il nome dei relativi *tag XML*)

7.2. Approccio Metodologico per SPCoop

L'approccio che viene suggerito in SPCoop, sulla base del contesto di riferimento precedentemente illustrato, è il seguente:

- Esistono moltissime metriche di risorsa e composte, algoritmi per la definizione dei parametri SLA a partire da essi, ecc. [TAPAS], e moltissime sono definibili in contesti specifici; tuttavia non tutte in generale sono applicabili al contesto SPCoop, in quanto richiederebbero un'eccessiva invasività nei confronti degli applicativi software che realizzano i servizi e non si limiterebbero a considerare solamente la Porta di Dominio, ledendo l'autonomia del singolo dominio. Si suggeriscono qui un insieme di parametri da considerare in una prima fase dell'operatività di SPCoop, da arricchire nel tempo con nuove definizioni, mano a mano che sempre più domini entreranno nel sistema SPCoop e forniranno (i) riscontri basati sulla loro esperienza nell'utilizzo delle definizioni e (ii) proposte di parametri considerati utili:
 - Durata del servizio (*service duration*), misurata come intervallo temporale in cui il servizio dovrebbe essere disponibile da parte dell'erogatore (ad es., "il servizio è disponibile ogni giorno lavorativo dalle 9.00 alle 12.00");
 - Tempo di servizio (*service time*), misurato come intervallo medio che intercorre tra la richiesta del servizio e l'effettivo espletamento (ad es., "la richiesta di servizio richiede 180 secondi per essere evasa");
 - Frequenza di servizio (*service rate*), misurata come numero di richieste di servizio che possono essere inoltrate all'erogatore in un intervallo temporale (ad es., "possono essere inoltrate 4000 richieste di servizio al giorno");
 - Tempo tra due fallimenti (*mtbf*), misurato come intervallo tra due momenti in cui il servizio non è disponibile (ad es., "il servizio è indisponibile ogni 300 giorni");

- Tempo di recupero (*recovery time*), misurato come tempo necessario a ripristinare il servizio quando questo è indisponibile (ad es., “una volta indisponibile, occorrono 24 ore per ripristinare il servizio”);
- Finestra di manutenzione (*maintenance window*), misurata come intervallo temporale in cui il servizio dovrebbe essere indisponibile per manutenzione (ad es., “il servizio non è disponibile il primo lunedì di ogni mese per manutenzione”);
- Disponibilità per regime temporale (*availability for varying time regimes*), misurata come percentuale di tempo in cui il servizio è disponibile in una data fascia temporale (ad es., “la disponibilità è del 99% nella fascia 9.00 – 12.00, del 60% durante la notte e nei weekend, ecc.”);
- Percentuale di conversazioni¹⁴ completate nel rispetto di determinati livelli di prestazioni (*percentage of conversations completed within defined performance levels*), misurata come percentuale (ad es., “una conversazione è completata nel 99% dei casi in 3 minuti, nel 0,15% in 1 ora e nel 0,05% in 3 ore”);

Ognuno di questi parametri deve essere definito come valore medio su opportune finestre temporali, in cui le caratteristiche della media e della finestra temporale variano in generale da un AS all'altro. Inoltre tali definizioni assumono un significato solo se contestualizzate secondo le specifiche condizioni in cui devono essere misurate. Ad esempio, si supponga di considerare il parametro tempo di servizio, e di considerare una finestra temporale di 200 secondi. Tale tempo di servizio non dovrebbe essere calcolato semplicemente come media sul numero di richieste di servizio che sono pervenute nei 200 secondi di osservazione, ma andrebbe specificata anche una condizione, ad es. che la misurazione è valida solo se nei 200 secondi non sono pervenute più di x richieste di servizio, in quanto per x molto grandi il servizio è in sovraccarico e questo non è permesso al fruitore (od almeno non si garantiscono più determinate prestazioni se il fruitore sovraccarica il servizio).

- Gli oggetti (*ServiceObject* nell'esempio con WSLA) su cui definire i parametri devono corrispondere alle operazioni definite nel WSDL logico dell'erogatore, poiché queste sono le effettive operazioni SOAP offerte dal servizio. Alcuni parametri però possono essere calcolati solamente facendo riferimento anche alle operazioni duali sul Web Service del fruitore (si consideri ad esempio il caso di una richiesta/risposta di tipo asincrono). In questi casi il lato fruitore deve essere considerato come una ulteriore strumentazione, che fornisce metriche di risorsa opportune per permettere il calcolo del parametro in esame.
- Affinché le metriche di risorsa, a partire dalle quali calcolare metriche composte e parametri SLA, siano disponibili, è necessario che le Porte di Dominio dell'erogatore e fruitore offrano dei meccanismi attraverso cui recuperare gli opportuni valori. Sono possibili due approcci:

¹⁴ Il lettore ricordi che una conversazione è una sequenza di operazioni che portano il servizio in uno stato in cui l'interazione con esso da parte del cliente può essere terminata, ed il cliente è in uno stato in cui ha ottenuto un risultato di valore. Pertanto considerare le conversazioni equivale a considerare interazioni significative/di valore con il cliente (e quindi considerare anche la prestazione rapportata all'efficacia del servizio). La stessa metrica può essere considerata sulla singola operazione del servizio, ma in questo caso non è detto che la singola operazione sia di valore per il cliente, e pertanto la metrica considererebbe la prestazione rapportata alla pura efficienza del servizio.

- Approccio *off-line*: la Porta di Dominio mantiene una tracciatura di tutti i valori delle metriche di risorsa che sono stati definiti nell'Accordo di Servizio; sulla base di tali registrazioni, è possibile a posteriori calcolare i parametri e verificare se sono stati rispettati gli SLO. Questa verifica viene eseguita periodicamente, ma in modo indipendente (*off-line*) rispetto all'esecuzione del servizio.

In tal caso la Porta di Dominio non deve offrire interfacce applicative dedicate all'accesso alle metriche di risorsa, ma è sufficiente che la sua implementazione interna preveda la tracciatura delle metriche. Tale tracciatura deve essere tale che, per ognuno dei punti temporali definiti nella direttiva di misurazione della metrica, sia disponibile la registrazione del valore assunto dalla metrica in esame.

Ad esempio, al fine di supportare l'approccio *off-line*, è necessario che WSLA venga esteso con la seguente definizione:

```
<xsd:complexType name="OFF-LINE">
  <xsd:complexContent>
    <xsd:extension base="wsa:MeasurementDirectiveType">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

- Approccio *on-line*: la Porta di Dominio offre delle interfacce applicative (a loro volta offerte come Web Service) attraverso cui è possibile, in modo concorrente (*on-line*) rispetto all'esecuzione del servizio, recuperare le metriche di risorsa. Sulla base di tali interfacce, opportuni servizi di misurazione e di valutazione delle condizioni (confrontare Figura 10) sono in grado di calcolare metriche composte, parametri SLA e valutare il rispetto degli SLO.

E' ipotizzabile che in una prima fase di SPCoop, le Porte di Dominio realizzino in modo obbligatorio il solo approccio *off-line*, ed opzionalmente l'approccio *on-line*. Successive evoluzioni di SPCoop, e la maggiore consapevolezza delle implicazioni e dei benefici di una valutazione e di un monitoraggio *on-line* dei livelli di servizio, porteranno nel prossimo futuro all'obbligatorietà dell'approccio *on-line*, ed alla definizione standardizzata di tali interfacce applicative per l'accesso alle metriche di risorsa.

- Non viene prevista in SPCoop la gestione automatica delle violazioni¹⁵. Viceversa, la presenza o meno di parti di supporto è una scelta delle singole coppie <erogatore, fruitore>.

A titolo di esempio, si consideri il servizio utilizzato nelle sezioni precedenti, e si supponga che un livello di servizio stabilisca che il tempo medio di risposta dell'operazione di richiesta/risposta asincrona debba essere inferiore a 15 secondi, valutato su una serie temporale (di durata un giorno) campionata ogni 30 secondi. Si assume che il calcolo venga eseguito in modalità *off-line* dall'erogatore. Allora il relativo frammento di un eventuale documento WSLA è riportato nel seguito:

```
<SLA xmlns=http://www.ibm.com/wsla
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xsi:schemaLocation="...">
  <Parties>
    <ServiceProvider name="EROGATORE">
      <Contact>
        <Street>...</Street>
        <City>...</City>
      </Contact>
```

¹⁵ Quindi nel caso dell'esempio con WSLA non sono presenti le specifiche delle ActionGuarantee.

```

</ServiceProvider>
<ServiceConsumer name="FRUITORE">
  <Contact>
    <Street>...</Street>
    <City>...</City>
  </Contact>
</ServiceConsumer>
</Parties>
<ServiceDefinition>
  <Operation name="RichiestaRispostaAsincrona_A"
    xsi:type="WSDLSOAPOperationDescriptionType">
    <Schedule name="MainSchedule">
      <Period>
        <Start>2005-02-28T09:00:00.000+01:00</Start>
        <End>2005-03-01T09:00:00.000+01:00</End>
      </Period>
      <Interval>
        <Seconds>45</Seconds>
      </Interval>
    </Schedule>
    <SLAParameter name="TempoRispostaMedio" type="float" unit="seconds">
      <Metric>tempoRispostaMedio</Metric>
    </SLAParameter>
    <Metric name="timestampRichiesta@Erogatore" type="time">
      <Source>EROGATORE</Source>
      <MeasurementDirective xsi:type="OFF-LINE"/>
    </Metric>
    <Metric name="timestampRisposta@Fruitore" type="time">
      <Source>FRUITORE</Source>
      <MeasurementDirective xsi:type="OFF-LINE"/>
    </Metric>
    <Metric name="tempoRisposta" type="float">
      <Source>EROGATORE</Source>
      <Function xsi:type="Minus" resultType="integer">
        <Operand>
          <Metric>timestampRisposta@Fruitore</Metric>
        </Operand>
        <Operand>
          <Metric>timestampRichiesta@Erogatore</Metric>
        </Operand>
      </Function>
    </Metric>
    <Metric name="tempoRispostaTS">
      <Source>EROGATORE</Source>
      <Function xsi:type="TSConstructor" resultType="TS">
        <Schedule>MainSchedule</Schedule>
        <Metric>tempoRisposta</Metric>
      </Function>
    </Metric>
    <Metric name="tempoRispostaMedio" type="float" unit="seconds">
      <Source>EROGATORE</Source>
      <Function xsi:type="Mean" resultType="float">
        <Metric>tempoRispostaTS</Metric>
      </Function>
    </Metric>
    <WSDLFile>ErogatoreLogico.wsdl</WSDLFile>
    <SOAPBindingName>...</SOAPBindingName>
    <SOAPOperationName>RichiestaRispostaAsincrona_A</SOAPOperationName>
  </Operation>
  ... ..
</ServiceDefinition>

<Obligations>
  <ServiceLevelObjective>

```

```
<Obligated>EROGATORE</Obligated>
<Validity>
  <Start>2005-02-28T09:00:00.000+01:00</Start>
  <End>2005-12-31T23:59:59.000+01:00</End>
</Validity>
<Expression>
  <Predicate xsi:type="LessEqual">
    <SLAParameter>TempoRispostaMedio</SLAParameter>
    <Value>15</Value>
  </Predicate>
</Expression>
<EvaluationEvent>NewValue</EvaluationEvent>
</ServiceLevelObjective>
... ..
</Obligations>
</SLA>
```

8. SPECIFICA DELLE CARATTERISTICHE DI SICUREZZA

Come discusso negli altri documenti SPCoop, la sicurezza del servizio è un argomento complesso e con molte sfaccettature. Chiaramente, la sicurezza in ambito Web Service è stata storicamente affrontata solo dopo la maturazione degli aspetti di base (definizione dei meccanismi di comunicazione, definizione dei meccanismi di descrizione delle interfacce, ecc.), e pertanto solo recentemente essa è stata oggetto di intensa attività di ricerca, sviluppo e standardizzazione; il risultato attuale è che sono presenti molte proposte interessanti, non ancora stabili e standardizzate, e che con molta probabilità subiranno delle modifiche nel corso del prossimo futuro. Pertanto, allo stato attuale, la definizione delle caratteristiche di sicurezza non può essere formalizzata, ma deve limitarsi a considerare possibili proposte alle quale si può ragionevolmente ricorrere. Successive evoluzioni delle regole per l'Accordo di Servizio, mano a mano che le tecnologie su questo specifico argomento evolveranno, progressivamente formalizzeranno anche questo elemento dell'AS.

Per trattare della sicurezza dei Web Service, bisogna considerare tutti i suoi aspetti:

- Identificazione (*identification*): stabilire univocamente chi sta richiedendo il servizio;
- Autenticazione (*authentication*): stabilire senza dubbio che chi sta richiedendo il servizio afferma di essere è vero.
- Autorizzazione (*authorization*): stabilire se il fruitore è autorizzato a richiedere lo specifico servizio.
- Integrità (*integrity*): garantire che non ci siano state manomissioni nelle informazioni trasmesse.
- Confidenzialità (*confidentiality*): garantire che solo erogatore e fruitore del servizio abbiano visibilità sulle informazioni trasmesse.
- Tracciamento (*audit*): mettere in atto meccanismi che permettano di tracciare qualsiasi evento legato all'erogazione/fruizione del servizio, in maniera da poter analizzare tali tracce per scoprire eventuali problemi.
- Non ripudio (*non-repudiation*): mettere in atto meccanismi per poter provare che l'erogatore abbia effettivamente erogato il servizio ed il fruitore lo abbia richiesto e poi fruito.

In generale, la sicurezza dei Web Service non può limitarsi al solo livello del protocollo di trasporto (HTTPS/SSL), in quanto esistono forti limitazioni delle garanzie su tutte le caratteristiche precedentemente menzionate. Al contrario, un'iniziativa estremamente interessante, a cui ricorrere per definire questo elemento dell'AS, è il Web Service Security Framework, che include una serie di specifiche (molte delle quali ancora allo stato di proposte) per affrontare problematiche di autenticazione, autorizzazione, privacy e confidenzialità, integrità, canali di comunicazione sicuri, tracciamento e la tematica più generale del trust tra differenti organizzazioni che cooperano. Poiché tale framework prevede come elementi di base solamente SOAP e WSDL, esso è perfettamente compatibile con le regole per l'AS fin qui presentate, e sembra ragionevole supporre che la sua maturazione e standardizzazione fornirà la

base tecnologica per la formalizzazione dell'elemento "Specifiche delle Caratteristiche di Sicurezza".

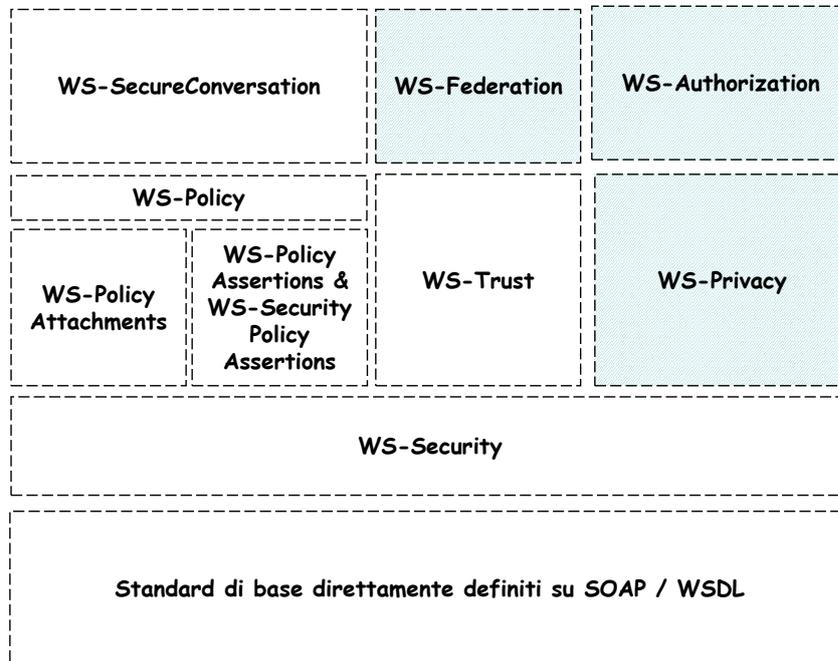


Figura 12. Elementi del Web Service Security Framework

In Figura 12 sono mostrate le tecnologie che vengono attualmente proposte per la sicurezza nei Web Service.

Trasporto Sicuro. Ai livelli più bassi sono definiti una serie di standard (alcuni direttamente definiti a partire da SOAP/WSDL) il cui obiettivo è fornire l'astrazione di "trasporto sicuro e confidenziale", tra i quali i più significativi:

- XML Digital Signature per firmare porzioni di documenti XML (e quindi (parte delle) informazioni scambiate tra erogatore e fruitore);
- XML Encryption per cifrare porzioni di documenti XML;
- SAML (Security Assertions Markup Language) per definire asserzioni relative a documenti, la loro provenienza ed i destinatari;
- XML Key Management Services, per definire l'interfaccia Web Service di servizi infrastrutturali preposti al processamento e gestione di chiavi crittografiche basate su PKI;
- WS-Security, che fornisce integrità e confidenzialità dei messaggi end-to-end, ovvero nel caso in cui il messaggio tra erogatore/fruitore attraversi differenti nodi intermedi.

Descrizione delle Caratteristiche. L'insieme di standard che ruotano intorno a WS-Policy permettono la descrizione, da parte di un Web Service, dei suoi requisiti e capacità, in modo tale che colui che invoca il Web Service ha le informazioni necessarie. In particolare, WS-PolicyAttachment permette di collegare queste asserzioni ad un Web Service, WS-

PolicyAssertions e WS-SecurityPolicyAssertions sono i linguaggi con cui effettivamente definire tali asserzioni.

Trust. WS-Trust definisce un modello per stabilire relazioni di trust (*fiducia*) tra soggetti (in particolare erogatore e fruitore), garantite da terze parti (indicate come Security Token Service), da realizzare come servizi infrastrutturali; vengono quindi definiti i protocolli e le interfacce per verificare con tali terze parti l'autenticità e la freschezza dei token dichiarati dai soggetti

Sicurezza della Conversazione. WS-SecureConversation definisce dei meccanismi per stabilire e scambiare dei contesti di sicurezza, da poter utilizzare per scambiare più messaggi facenti parte di una stessa conversazione.

Queste proposte sono poi complementate da altre, in particolare WS-Privacy, WS-Authorization e WS-Federation, che tuttavia sono ancora ad uno stadio molto primordiale (e quindi sono colorate tratteggiate in Figura 13).

In base queste tecnologie/proposte, l'elemento di Specifica delle Caratteristiche di Sicurezza di un AS sarà composto da una serie di documenti XML (conformi alle specifiche appena citate) che descrivono (i) quali tecnologie vengono utilizzate, e come, per avere l'astrazione di canale sicuro e confidenziale tra erogatore e fruitore, (ii) le caratteristiche ed i requisiti di sicurezza presentati dal servizio e (iii) eventuali contesti di sicurezza utilizzati nelle conversazioni supportate dal servizio. La realizzazione del servizio farà poi uso di meccanismi di trust per verificare i token di sicurezza, attraverso l'invocazione di opportuni servizi infrastrutturali (a loro volta offerti come Web Service) che dovranno essere offerti da SPCoop.

9. DOMINIO DI COOPERAZIONE, ACCORDO DI COOPERAZIONE E RELAZIONI CON L'ACCORDO DI SERVIZIO

Dal un punto di vista astratto, un Accordo di Servizio descrive una comunicazione/collaborazione *2-party*, in cui un soggetto offre un servizio applicativo SPCoop ed un altro soggetto fruisce di tale servizio. Pertanto, tutti i servizi applicativi offerti da un Dominio rientrano nella responsabilità del singolo soggetto.

In realtà molti procedimenti e compiti istituzionali non riguardano l'operato di una singola amministrazione, ma vedono altresì il concorso dell'azione di più soggetti. Tale situazione, che in base ai processi organizzativi di decentramento e di delega dallo Stato centrale verso le Regioni e gli Enti locali è sempre più frequente, si esemplifica in due principali tipologie di interazione:

- procedimenti inter-amministrativi, nei quali più amministrazioni concorrono, con compiti diversi, al conseguimento di un risultato complessivo, riconducibile ad uno o più servizi integrati erogati sia a fruitori esterni alla PA (ad es., Sportello unico alle imprese, Sportello unico per l'immigrazione, ecc.) che a fruitori interni alla PA. Questo tipo di procedimenti sono incentrati sulla amministrazione che eroga il servizio integrato finale;
- procedimenti di razionalizzazione, coordinamento e controllo, in cui è individuato normativamente un soggetto vigilante e/o di regolazione, a livello centrale o territoriale (ad es., Regioni, Province, ecc.), mentre le funzioni amministrative sono attribuite a soggetti periferici, tipicamente enti locali (ad es., Anagrafi, Catasto, Demanio...) che erogano sul territorio una stessa gamma di servizi.

Il Dominio di Cooperazione rappresenta la formalizzazione della volontà di associazione tra diversi soggetti per cooperare nella informatizzazione di un insieme di procedimenti amministrativi pertinenti. Nel Dominio di Cooperazione deve essere individuato un *soggetto coordinatore responsabile*, che assicura l'efficacia organizzativa e tecnica della cooperazione ed il coordinamento degli adempimenti di ciascuno dei soggetti partecipanti, e l'insieme dei *servizi applicativi composti* che il Dominio di Cooperazione eroga verso l'esterno. Infatti esternamente il Dominio di Cooperazione è un erogatore di servizi al pari dei Domini di responsabilità delle singole amministrazioni; la differenza è nella realizzazione di tali servizi, che nel caso del Dominio di Cooperazione nascono dall'integrazione e composizione dei servizi offerti dai singoli domini regolata sulla base di accordi specifici tra le parti in causa, mentre nel caso del singolo dominio la realizzazione si appoggia ad applicazioni completamente sotto la responsabilità della singola amministrazione.

Il Dominio di Cooperazione deriva spesso dalla trasposizione di una norma di legge, dove sono individuati i soggetti coinvolti e quello che ha responsabilità del coordinamento o della vigilanza. Ove i ruoli non siano esplicitamente previsti dalla normativa, vale il principio generale di pariteticità delle amministrazioni che, in tal caso, concordano quale tra di loro è responsabile del Dominio di Cooperazione. Il principio di titolarità (e quindi di responsabilità) dell'azione amministrativa sul singolo adempimento/procedimento è comunque valido. Ne

consegue che nel Dominio di Cooperazione ciascun adempimento o parte di procedimento è associato al soggetto pubblico che istituzionalmente ne ha la responsabilità. Lo stesso rimane responsabile dei dati e dei servizi scambiati di cui è normativamente titolare.

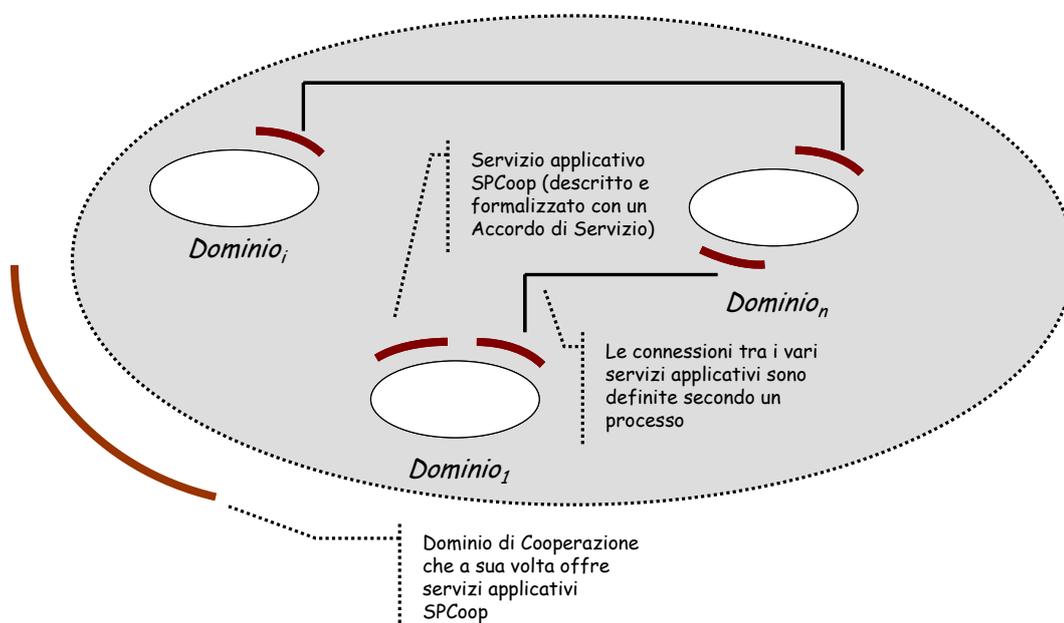


Figura 14. Dominio di Cooperazione

Un **Accordo di Cooperazione (ACoop)** è in sintesi la specifica dei servizi applicativi offerti da un Dominio di Cooperazione. Come evidenziato in Figura 14, tre sono gli elementi fondamentali che caratterizzano l'erogazione di servizi applicativi da parte di un Dominio di Cooperazione:

- i servizi applicativi che il Dominio di Cooperazione offre all'esterno. Dal punto di vista del fruitore, questi sono indistinguibili da qualsiasi altro servizio applicativo offerto direttamente da un Dominio, e vengono descritti attraverso un Accordo di Servizio; nel seguito verranno indicati come *servizi composti*;
- i servizi applicativi che il Dominio di Cooperazione utilizza internamente, componendoli, per offrire i servizi composti; anche questi sono descritti attraverso il proprio Accordo di Servizio; nel seguito verranno indicati come *servizi componenti*;
- per ognuno dei servizi applicativi composti, la specifica della modalità secondo cui i servizi componenti sono coordinati al fine di offrire il servizio composto. Tale specifica può essere definita secondo due diverse ottiche:
 - dal punto di vista interno al servizio composto, ovvero descrivendo il processo secondo cui i servizi componenti devono essere coordinati per offrire il servizio composto. In tal caso si parla di *orchestrazione*;
 - dal punto di vista esterno, ovvero descrivendo i vincoli sugli scambi di messaggi tra i vari servizi componenti; in tal caso si ha cioè la specifica di una collaborazione *N-party*. Si parla allora di *coreografia*.

La descrizione di un'orchestrazione non può prescindere da quella della coreografia. Se infatti si hanno $N + 1$ servizi, di cui uno è quello composto, con l'orchestrazione viene descritto come si realizza appunto il servizio composto in base agli altri N , mentre con la coreografia vengono descritti gli scambi tra gli $N + 1$ servizi; ma è chiaro che la realizzazione di un servizio (in particolare di quello composto) è influenzata dalla coreografia nella quale esso è incluso, e viceversa la coreografia a cui può partecipare un servizio dipende dal suo processo (composto a partire dagli altri N).

Attualmente esiste uno standard maturo per la descrizione di un'orchestrazione, in particolare WS-BPEL (Web Service Business Process Execution Language [WS-BPEL]), e sono in corso delle iniziative di standardizzazione della coreografia, in particolare WS-CDL (Web Service Choreography Description Language [WS-CDL]).

Pertanto un Accordo di Cooperazione si compone di:

- un *documento istitutivo* in linguaggio naturale che descrive le finalità ed il fondamento normativo o istituzionale del Dominio di Cooperazione;
- un insieme di riferimenti ordinati agli Accordi di Servizio che descrivono i servizi composti offerti dal Dominio di Cooperazione;
- un insieme di documenti WS-BPEL che descrivono il processo di coordinamento tra i servizi composti¹⁶; tali documenti possono anche servire per l'esecuzione diretta, attraverso opportune tecnologie di orchestrazione che interpretano dinamicamente i documenti WS-BPEL, del servizio composto da parte dell'organizzazione responsabile;
- un insieme di liste di riferimenti agli Accordi di Servizio che descrivono i servizi componenti.

In termini formali¹⁷ si ha:

$$\begin{aligned}
 \text{ACoop} = & \langle \\
 & \text{Documento,} \\
 & \{ \text{AS}_1, \dots, \text{AS}_i, \dots, \text{AS}_n \}, \\
 & \{ \text{O}_1, \dots, \text{O}_i, \dots, \text{O}_n \}, \\
 & \{ \text{AS}_1^1, \dots, \text{AS}_j^1, \dots, \text{AS}_m^1, \\
 & \quad \dots \\
 & \quad \text{AS}_1^i, \dots, \text{AS}_j^i, \dots, \text{AS}_m^i, \\
 & \quad \dots \\
 & \quad \text{AS}_1^n, \dots, \text{AS}_j^n, \dots, \text{AS}_m^n \} \\
 & \rangle
 \end{aligned}$$

¹⁶ Future evoluzioni dell'Accordo di Cooperazione prenderanno in considerazione anche la specifica di altrettanti documenti per la descrizione della coreografia.

¹⁷ Si indica con AS_i l'accordo di servizio dell' i -esimo servizio composto offerto dal Dominio di Cooperazione, con O_i il documento che descrive la i -esima composizione che realizza il servizio composto a partire da m servizi componenti, e con AS_j^i l'accordo di servizio del j -esimo servizio componente che viene composto per realizzare l' i -esimo servizio composto.

da cui emerge che il servizio composto descritto dall'Accordo di Servizio AS_i , si compone a partire dall'insieme dei servizi descritti da altrettanti Accordi di Servizio $\{AS_1^i, \dots, AS_m^i\}$, secondo il processo descritto dall'orchestrazione O_i .