

(Titolo proposto)

**Le ontologie come infrastruttura concettuale per
lo sviluppo dei servizi assistiti da tecnologie
semantiche nella Pubblica Amministrazione**

**Gruppo di Lavoro per la standardizzazione dei metadati
e della semantica dei servizi applicativi in rete
nella pubblica amministrazione¹**

Editor: Aldo Gangemi^{1,2}, Stefano Fuligni¹, Francesco Tortorelli¹

¹*CNIPA, Roma*, ²*ISTC-CNR, Roma*

9 novembre 2007

Partecipanti che hanno contribuito il materiale su cui si basa il rapporto tecnico:

Annalisa Barone³, Flavio Corradini⁴, Paolo Di Pietro³,
Roberta Ferrario⁸, Stefano Fuligni¹, Aldo Gangemi²,
Riccardo Grosso⁷, Maurizio Lenzerini⁵, Marco Pistore⁶,
Barbara Re⁴, Francesco Tortorelli¹, Michele Trainotti⁶,
Guido Vetere⁹, ...

¹*CNIPA, Roma*, ²*ISTC-CNR, Roma*

³*Diviana, Roma*, ⁴*Università di Camerino*,

⁵*Università di Roma I*, ⁶*Università di Trento*,

⁷*CSI Piemonte*, ⁸*ISTC-CNR, Trento*, ⁹*IBM Research, ...*

Partecipanti del Gruppo di Lavoro:

Tabella 1: Lista dei partecipanti al Gruppo di Lavoro

Andrea Valle	Adobe
Giuseppe Neri	AITech-Assinform
Marco Bottone	Atos Origin
Massimo Anelli	Atos Origin
Andrea Nicolini	CISIS
Fabio Vernacotola	CM Sistemi
Alessandro Alessandroni	CNIPA
Alfio Raia	CNIPA
Carlo Galli	CNIPA
Caterina Lupo	CNIPA
Elettra Cappadozzi	CNIPA
Francesco Tortorelli	CNIPA
Stefano Fuligni	CNIPA
Paolo Luxardo	CONSIP
Andrea Muraca	CSI-Piemonte
Lucio Micheli	EDS
Stefano Bruschini	EDS
Epifanio Giannetto	FINSIEL
Marco Neri	FINSIEL
Guido Vetere	IBM
Amedeo Bogliaccino	INPS
Annarita Sala	INPS
Francesco Sasso	INSIEL
Lina Lolli	INSIEL
Marco Pistore	ITC Trento
Paolo Traverso	ITC Trento
Aldo Gangemi	ISTC-CNR, CNIPA
Nicola Guarino	ISTC-CNR
Flavio Iodice	Ministero del Lavoro
Luca Torri	Ministero del Lavoro (Italiavoro)
Lidia Di Minco	Ministero della Salute
Pierpaolo Benintende	Oracle Italia
Nicola Prantil	Provincia Autonoma di Trento
Lucio Forastieri	Regione Marche
Marco Giannotti	SAS
Danilo Poccia	SUN
Barbara Re	Università di Camerino
Flavio Corradini	Università di Camerino
Maurizio Lenzerini	Università di Roma La Sapienza
Paolo Di Pietro	Diviana
Annalisa Barone	Diviana
Roberta Ferrario	ISTC-CNR
Riccardo Grosso	CSI-Piemonte
Michele Trainotti	ITC Trento
Grazia Strano	Ministero del Lavoro

Indice

1	Introduzione	4
1.1	Obiettivi del Gruppo di Lavoro e contesto di riferimento	4
1.2	Struttura del documento	4
2	La semantica a supporto dell'interoperabilità dei servizi	5
2.1	Modelli per l'interoperabilità semantica nelle architetture orientate ai servizi	5
2.1.1	Abstract	5
2.1.2	Introduzione	5
2.1.3	Lo stato dell'arte	7
2.1.4	La semantica delle infrastrutture di servizi	8
2.1.5	La semantica FOL	9
2.1.6	La semantica modale	11
2.1.7	Il ruolo delle ontologie	13
2.1.8	Modelli di interoperabilità semantica	14
2.1.9	Modello any-to-any centralizzato	15
2.1.10	Modello any-to-one centralizzato	17
2.1.11	Modello any-to-any decentralizzato	20
2.1.12	Modello any-to-one decentralizzato	22
2.1.13	In conclusione	23
2.1.14	Note	24
2.2	e-Government: Problemi di Interoperabilità Semantica in un Contesto Distribuito	27
3	Linee-guida e componenti per lo sviluppo e l'uso delle ontologie nella PA	29
3.1	Cosa sono le ontologie informatiche e come si progettano	29
3.1.1	Introduzione	29
3.1.2	L'ingegneria ontologica	34
3.1.3	I linguaggi per la rappresentazione delle ontologie	35
3.1.4	I contenuti delle ontologie	40
3.1.5	Apprendimento di ontologie da testi	46
3.1.6	Qualità di un'ontologia	50
3.1.7	Strumenti software	51

3.1.8	Lecture consigliate	52
3.2	Esempi di ontologie riusabili e come impiegarle per l'integrazione semantica dei domini della PA	54
3.2.1	Introduzione	54
3.2.2	Domande di competenza, task e contesto dell'ontologia	54
3.2.3	Selezione delle risorse disponibili e metamodelli di reingegnerizzazione	57
3.2.4	Ontologie fondazionali e di riferimento	57
3.2.5	Design guidato da modelli riusabili	57
3.2.6	Regole di mappatura e interoperabilità fra ontologie	57
3.3	Un'ontologia fondazionale per l'interoperabilità semantica nell'e-Government	58
3.3.1	Introduzione	58
3.3.2	Obiettivi della proposta	59
3.3.3	Stato dell'arte e risultati preliminari	60
3.3.4	Rilevanza della proposta per SPCoop	62
3.3.5	Riferimenti bibliografici	64
3.4	Un'ontologia di riferimento per la PA	66
3.4.1	Introduzione	66
3.4.2	Architettura modulare	66
3.4.3	Struttura interna	68
3.5	SAWSDL e la semantica dei servizi	80
3.6	Annotazione semantica di Servizi Web	80
3.6.1	Introduzione	80
3.6.2	Interoperabilità e Servizi Web	81
3.6.3	Semantica e Servizi Web	83
3.6.4	Conclusioni	94
3.7	Ontologie, servizi e metadati	95
3.8	Metadati	95
3.8.1	Introduzione	95
3.8.2	Modelli di Metadati	96
3.8.3	Caratteristiche Utilizzi e Funzioni dei Metadati	99
3.8.4	Tool per la gestione dei metadati	100
3.8.5	Metadati soluzioni a supporto della Cooperazione Applicativa SPCoop	101
3.8.6	Metadati e soluzioni di e-Government	102
3.8.7	Architettura dei Dati: RDF	102
3.8.8	Conclusioni	104
4	Scenari applicativi per la semantica nella PA	106
4.1	Confronto fra scenari applicativi nella PA	106
4.2	Il catalogo metadati della PA piemontese	107
4.2.1	Il catalogo metadati della Pubblica Amministrazione Locale piemontese	107
4.2.2	Esperimenti di estrazione e correlazione di concetti	109

4.3	Sviluppo di un'ontologia per l'eGovernment: un caso di studio italiano	114
4.3.1	Abstract	114
4.3.2	Introduzione	114
4.3.3	Il documento "Atto di Nascita"	115
4.3.4	Descrizione dell'ontologia	116
4.3.5	Descrizione delle Clausole	121
4.3.6	Conclusioni	122
4.3.7	Riferimenti bibliografici	124
4.4	A case study of semantic solutions for citizen-centered Web portals in eGovernment: the Tecut Portal	125
4.4.1	Abstract	125
4.4.2	Introduction	125
4.4.3	State of the Art	126
4.4.4	The Tecut Portal	127
4.4.5	LifeEvents as organizer	129
4.4.6	Semantic Life Events	132
4.4.7	Conclusions and Future works	134
4.4.8	References	135
4.5	Un'esperienza di modellazione dei servizi di e-Government	138
4.5.1	Introduzione	138
4.5.2	L'Ontologia della Pubblica Amministrazione Locale	140
4.5.3	Architettura: un modello di riferimento	140
4.5.4	Pattern di Interoperabilità	140
4.5.5	L'approccio basato sul riuso	141
4.5.6	La Base di Conoscenza UML	141
4.5.7	Pubblicazione della Base di Conoscenza	151
4.5.8	Benefici derivanti da questo approccio	151
4.5.9	Alcuni numeri	152
4.5.10	Prossimi passi	152
5	Caso di studio: un'ontologia per il mondo del lavoro	155
5.1	Integrazione di modelli XSD per la PA che descrivono il mondo del lavoro	155
5.1.1	Introduzione	155
5.1.2	Requisiti di INPS e BCNL	155
5.1.3	Il modello dell'INPS per le "Comunicazioni obbligatorie"	155
5.1.4	Il modello della BCNL	155
5.1.5	Il lavoratore in Arianna	155
5.1.6	Il lavoratore nei modelli internazionali	155
5.1.7	Problemi	155
5.2	Proposta di metodologia per l'annotazione semantica di accordi di servizio	156
5.2.1	Introduzione	156
5.2.2	Accordo di Servizio	157
5.2.3	Semantica	158

5.2.4	Strumenti	162
5.2.5	Un esempio pratico	163
5.2.6	Appendice A	165
5.3	Integrazione dei modelli usando l'ontologia di riferimento	170
6	Gestione degli aspetti semantici dei servizi nella PA	171
6.1	Servizi di gestione della semantica nella PA	171
6.2	Esperienze e linee-guida per la formazione da parte del CNIPA, della ricerca e delle aziende	171

Capitolo 1

Introduzione

1.1 Obiettivi del Gruppo di Lavoro e contesto di riferimento

TBD

1.2 Struttura del documento

Il rapporto è organizzato in sei capitoli: (i) introduzione, (ii) semantica e interoperabilità dei servizi, (iii) linee-guida e componenti per la progettazione di ontologie nella PA, (iv) scenari applicativi, (v) caso di studio nel dominio Lavoro, (vi) gestione aspetti semantici della PA.

Il rapporto è inteso per una lettura modulare, anche se la sequenza dei capitoli segue una narrativa classica: dalla teoria agli scenari attuali e all'applicazione. Ogni capitolo contiene contributi di più partecipanti al Gruppo di Lavoro, con raccordi e outline curati dall'editor.

Capitolo 2

La semantica a supporto dell'interoperabilità dei servizi

2.1 Modelli per l'interoperabilità semantica nelle architetture orientate ai servizi

Basato su materiale contribuito da:

Guido Vetere (IBM Research) e Maurizio Lenzerini (Università di Roma 1)

2.1.1 Abstract

Sebbene le architetture SOA abbiano compiuto enormi passi avanti verso l'implementazione dell'interoperabilità all'interno di ambienti eterogenei distribuiti come SPCoop, gestire le differenze semantiche in questi contesti costituisce tuttora una sfida. In questo capitolo presentiamo una panoramica del problema dell'interoperabilità semantica e trattiamo il ruolo delle ontologie. Procediamo quindi ad analizzare quattro modelli basilari di interoperabilità semantica che differiscono sia nella mappatura tra descrizioni dei servizi e ontologie, sia nel modo in cui viene effettuata la valutazione della logica di integrazione.

2.1.2 Introduzione

I moderni sistemi di Information Technology basati su architetture Service-Oriented sono composti da una rete di provider di servizi. I servizi vengono richiamati dalle applicazioni client (i consumatori) mediante messaggi conformi a specifici schemi descrittivi. Nonostante le descrizioni dei servizi vengano tipicamente esportate dai provider verso appositi registry (o directory), alcune di esse possono essere fornite direttamente ai consumatori, o essere fissate

mediante convenzioni esterne al sistema. Quello che è cruciale è che (a differenza delle architetture client/server) ogni servizio possiede uno schema pubblico che ne definisce l'interfaccia applicativa, e appositi registri rendono possibili la consultazione e il binding dei servizi senza che sia a priori necessaria alcuna conoscenza specifica di essi.

I servizi supportati da un provider danno accesso allo stato di quest'ultimo e consentono di apportare modifiche allo stato stesso. Qui utilizziamo il termine stato in modo classico, ovvero come valori effettivi assunti da un determinato insieme di attributi. Per esempio un servizio orientato ai dati può essere visto come un sistema di gestione database il cui stato è rappresentato dalle informazioni residenti nel database, e nel quale le applicazioni client possono effettuare operazioni di lettura e scrittura attraverso un insieme di servizi descritti in termini di operazioni di accesso e modifica ammesse, in cui lo schema del servizio include da una porzione dello schema del database esportato in maniera appropriata.

Affinché le applicazioni client possano usare i servizi in maniera efficiente, è essenziale che i progettisti delle applicazioni comprendano le descrizioni dei servizi corrispondenti alle operazioni e alle strutture dati del client. In modo simile, gli sviluppatori di sistemi informativi distribuiti che cooperano per mezzo di servizi dovrebbero possedere una buona comprensione delle interrelazioni tra le descrizioni dei servizi. Metodologie, soluzioni e tecniche che mirano alla corretta interpretazione e implementazione delle descrizioni dei servizi costituiscono quello che può essere chiamato livello semantico delle infrastrutture basate sui servizi.

In un ambiente service-oriented, un'appropriata gestione del livello semantico mira ad assicurare che i dati inseriti nei messaggi vengano interpretati da provider e consumatori in modo da denotare le stesse entità, proprietà e relazioni del mondo a cui il sistema di cooperazione inerisce. Per esempio, un layer semantico può modellare il fatto che l'attributo costo-del-tour del sito www.grand-tour.com corrisponda all'attributo prezzo-del-biglietto del sito www.railways.it. Il livello semantico riguarda dunque la modalità con cui i partecipanti possono interpretare le descrizioni e i dati del sistema rispetto ad una ontologia(1) del dominio business, e come questa interpretazione possa essere condivisa e resa trasparente in tutta l'infrastruttura.

Il livello semantico può anche comprendere anche complessi aspetti operativi, come ad esempio la definizione delle transazioni business prevista dagli standard RosettaNet(2). Infatti, esso può essere visto come contenitore di qualunque cosa che possa essere inserita in un vocabolario di dati allo scopo di caratterizzare il servizio del provider. In altre parole, il livello semantico riguarda oggetti, proprietà, eventi, stati e qualsiasi altra cosa concepibile, esprimibile e scambiabile attraverso una rete di comunicazione - cosa che in pratica può corrispondere alla copertura di un intero dizionario linguistico standard.

In conclusione, l'interesse per la semantica nelle architetture di servizi integrati come SPCop è motivato dalla necessità di fare in modo che sistemi informativi eterogenei possano funzionare insieme all'interno di un mondo concreto

di entità e processi, in quanto capaci di ragionare su rappresentazioni omogenee e integrate di tale mondo.

2.1.3 Lo stato dell'arte

Le infrastrutture per l'integrazione dei sistemi della Pubblica Amministrazione sono state per lungo tempo progettate sotto il profilo del 'trasporto' e del 'middleware'. Gli aspetti semantici della cooperazione applicativa sono stati inquadrati nell'ambito delle attività di disegno e implementazione dei singoli servizi, in cui le corrispondenze semantiche sono catturate da regole di trasformazione dei messaggi che mappano, caso per caso, simboli, valori e strutture nei messaggi scambiati. In generale questo approccio può sembrare semanticamente neutrale dato che non richiede che le mappature riflettano il modo in cui la fonte e la destinazione si riferiscono a entità reali. In altre parole, le regole di trasformazione dei messaggi sono scritte senza tener conto del motivo per cui le mappature corrispondenti sono concettualmente valide rispetto al dominio applicativo. Questa strategia può essere considerata una maniera pragmatica per affrontare l'interoperabilità semantica in molte situazioni comuni quando è ragionevole supporre che le interpretazioni date dai partecipanti alle descrizioni dei servizi siano omogenee. In molti casi questa neutralità presuppone una sorta di realismo: l'idea che gli attributi che conformano il dominio dell'infrastruttura di servizi siano assegnati in modo naturale per tutti e per sempre. Così, dato che i servizi utilizzano implicitamente la medesima ontologia, il ruolo dei progettisti è solo quello di neutralizzare un numero di differenti rappresentazioni nominali e strutturali degli elementi di un'unica concettualizzazione universale e immanente.

Restare neutrali rispetto alla semantica è ragionevole in molte situazioni, ma non è un atteggiamento valido in generale, e mostra particolari criticità in un contesto come quello della cooperazione applicativa nel Sistema Pubblico di Connettività. Infatti questo approccio presuppone l'affidabilità di un qualche accordo implicito esistente al di fuori dell'infrastruttura, per esempio nel contesto organizzativo e sociale che circonda il sistema. Per esempio, confidare sul fatto che due attributi siano riferiti alla medesima entità perché possiedono il medesimo nome si basa sul presupposto che le loro etichette vengano interpretate uniformemente da tutte le parti coinvolte. Questo potrebbe essere accettabile qualora le infrastrutture di servizi venissero costruite, implementate e gestite nel ristretto ambito di confini organizzativi ben definiti come accade negli scenari di integrazione applicativa aziendali, dove le policy relative ai nomi possono essere applicate e verificate. Ma SPCoop è la rete di una grande e complessa organizzazione come la Pubblica Amministrazione, nelle sue articolazioni centrali e locali, e nei suoi rapporti con i cittadini, le imprese, la società civile. I progettisti dei sistemi della PA possono ben difficilmente fare assunzioni sul contesto dell'infrastruttura, e dunque una derivazione accurata del significato dei servizi non può essere lasciata implicita. Pertanto, la gestione del livello semantico dell'SPCoop necessita di strumenti e metodi specifici.

L'approccio corrente alla cooperazione applicativa SPCoop recepisce l'esi-

genza di dotare il Sistema Pubblico di un'appropriata gestione della semantica dei dati e dei servizi della PA. Metodi e strumenti per questa gestione devono tenere in considerazione le diverse scelte architetturali che sono possibili nell'ambito delle soluzioni SOA. Nella prossima parte forniremo alcuni concetti basilari per comprendere l'interoperabilità semantica; quindi procederemo a classificare i modelli di interoperabilità semantica negli ambienti SOA tenendo conto se questi riflettano una struttura basata su hub o meno, e se siano o no basati su modelli di dominio specifici. Mostriamo anche come i problemi dell'interoperabilità semantica siano vicini a molti problemi studiati da decenni nel campo dello scambio e dell'integrazione di dati in ambito logico,⁽⁷⁾ e come alcuni concetti sviluppati in quest'area possano essere applicati vantaggiosamente quando si progettano nuove tipologie di infrastrutture IT.

2.1.4 La semantica delle infrastrutture di servizi

Nella sezione precedente abbiamo mostrato come i servizi permettano di consultare e modificare lo stato dei loro provider, ovvero i valori di alcuni attributi esportati, e come queste operazioni di accesso e manipolazione siano effettuate inviando e ricevendo, nella fattispecie, messaggi di e-Government. Abbiamo anche introdotto informalmente il layer semantico come un framework che consente un'interpretazione omogenea dei messaggi scambiati tra i servizi rispetto a una qualche concettualizzazione sottostante al sistema. Prenderemo ora in esame la semantica in maggior dettaglio partendo dalla caratterizzazione di un singolo servizio per poi considerare l'interoperabilità (o integrazione) dei servizi.

Informalmente, un servizio è un sistema dotato di un proprio stato interno (per esempio i valori residenti nell'istanza di un database) caratterizzato da uno schema interno (per esempio relazionale) che espone all'esterno un set di metodi di accesso e manipolazione per mezzo di un'interfaccia. L'interfaccia di un servizio si basa su due alfabeti di simboli - l'alfabeto operativo e l'alfabeto dei dati - e un set di primitive strutturali e connettivi logici che compongono il linguaggio descrittivo. Ciò fornisce ai progettisti i mezzi per formulare una caratterizzazione pubblica del servizio, quella che è chiamata descrizione. Per esempio, nella Figura 1 WSDL (Web Services Description Language) e XML (Extensible Markup Language) Schemas forniscono congiuntamente un linguaggio che informa sulla disponibilità di un'operazione chiamata 'getRoute' e sulla sua eventuale corrispondenza a una funzione che mappa le località rispetto ai percorsi. Descrizioni di questo tipo possono essere considerate come teorie che forniscono definizioni e vincoli per spiegare l'ontologia e il comportamento di un servizio. L'interfaccia è legata allo stato interno del servizio per mezzo di un'implementazione appropriata.

[Figura 1]

Il problema dell'integrazione semantica scaturisce dal fatto che, sebbene le descrizioni dei servizi siano pubbliche, le implementazioni dei servizi stessi non sono accessibili dall'esterno. Così, tanto per i client quanto per i provider, la corrispondenza tra descrizione del servizio e stato interno è essenzialmente opaca (ovvero nascosta). Consideriamo per esempio l'applicazione Grand Tour Agen-

cy della Figura 2, che utilizza i servizi forniti dallo European Travel Bureau che agisce da broker e che a sua volta utilizza i servizi forniti dalle Ferrovie Italiane. Per programmare un viaggio, Grand Tour Agency specifica il treno come mezzo di trasporto preferito e richiede allo European Travel Bureau il set di percorsi (e i prezzi ad essi associati) tra due destinazioni. Per soddisfare la richiesta, lo European Travel Bureau utilizza i servizi delle Ferrovie Italiane. In qualità di broker, lo European Travel Bureau deve essere in grado di interpretare le informazioni relative ai treni che percorrono la specifica tratta e che vengono fornite dalle Ferrovie Italiane; e in modo tale per cui la tariffa, così come intesa dal client, corrisponda al costo del biglietto ferroviario, così come inteso dal provider. Come mostrato nella Figura 2, per poter gestire una cooperazione di questo tipo i tre partecipanti esportano schemi e scambiano dati sulla base di mappature. Gli schemi esplicitano i predicati relazionali di ciascun servizio in notazione astratta, mentre le mappature rappresentano le implicazioni logiche tra questi predicati. Per indicare la possibilità di implementare queste mappature con una varietà di policy di trasferimento dati si usano frecce ondulate anziché le classiche frecce dritte. Da notare che, mentre tanto il consumatore quanto il provider possiedono sistemi interni (database, file, programmi) che implementano funzioni e strutture dati corrispondenti agli schemi esposti, il sistema del broker è soprattutto dedicato a gestire e valutare le mappature degli schemi.

[Figura 2]

Il client utilizza il servizio per ottenere i dati, elaborarli e quindi aggiornare i propri database; può inoltre inviare messaggi contenenti dati allo scopo di modificare lo stato del provider. I dati e il linguaggio operativo che descrive il servizio devono essere mappati rispetto alle strutture di dati e alle trasformazioni di stato presso il client. Per gli sviluppatori di Grand Tour Agency questo solleva due problemi semantici: produrre messaggi appropriati per il servizio (per prelevare o per modificare dati) e interpretare correttamente le informazioni restituite - cosa che richiede la comprensione del significato delle espressioni accettate e prodotte dal servizio e la sicurezza che tale significato sia omogeneo rispetto al modo in cui i dati forniti dal servizio vengono usati dall'applicazione client. Similmente, per gli sviluppatori delle Ferrovie Italiane il problema è quello di implementare lo schema del servizio interpretando il significato delle espressioni descrittive fornite dallo European Travel Bureau. Ma cosa significa realmente, per un sistema, capire il significato? A grandi linee ciò è correlato a quanto, in logica, è chiamato interpretazione. Per spiegare questo concetto introduciamo la semantica FOL (First-Order Logic, o logica del primo ordine).

2.1.5 La semantica FOL

Un'interpretazione FOL dell'interfaccia di un servizio è fondamentalmente una funzione che associa simboli del linguaggio dati a tuple di dati nello stato interno del provider del servizio (nota anche come estensione del simbolo(8)), mentre il significato di una descrizione è semplicemente un calcolo della sua estensione basata sui connettivi logici. facile vedere come ciò corrisponda alla semanti-

ca standard dei database. Per esempio, il significato dell'espressione id,tm — $Treno(id, Roma, Milano, tm)$ (ovvero un set di identificatori e orari di partenza sulla relazione $Treno$ con vincoli di partenza e destinazione) sarebbe - in un'interpretazione FOL nel dominio del provider di Figura 2 - un set $\{id, tm\}_1, \dots, \{id, tm\}_n$ di tuple nel database interno. Il modello di un servizio basato sulla sua interfaccia è un'interpretazione che soddisfa tutte le formule della descrizione del servizio rispetto allo stato del provider, e qualunque stato di un servizio - se consistente - è previsto che sia un modello in una qualche interpretazione appropriata. La semantica FOL è semplice: ci si può ragionevolmente attendere che i provider forniscano dati secondo i loro modelli, e ottenere l'interoperabilità semantica, in linea di principio, potrebbe richiedere solo che i consumatori consumino tali modelli tenendo correttamente in considerazione i vincoli imposti dagli schemi.

Sfortunatamente esistono numerose limitazioni all'impiego della semantica FOL nell'ambito di scenari loosely-coupled come l'interoperabilità dei Web Services. In primo luogo occorre notare che l'implementazione del provider non è visibile ai consumatori: ciò significa, per esempio, che l'estensione dei predicati (i simboli dell'alfabeto dati) può essere modificata dai provider. I client hanno ben poco controllo sul modo in cui queste estensioni variano nel tempo: non possono interrogare i loro provider troppo spesso, e quindi i modelli possono facilmente perdere la sincronizzazione. Consideriamo inoltre uno scenario di integrazione dati nel quale un client voglia popolare il proprio modello con dati provenienti da fonti differenti (per esempio due servizi che espongono la medesima interfaccia), e supponiamo che i modelli siano reciprocamente incompatibili (per esempio, un servizio afferma che il treno parte alle 13.00 quando l'altro invece indica le ore 14.00). Come è possibile determinare il modello client in una situazione del genere? Infine, pensiamo al tipico caso in cui un consumatore integri i dati provenienti da un provider all'interno del modello di una teoria più ampia comprendente predicati e vincoli attinti da altri servizi e assemblati per mezzo di mappature: come si può garantire che gli enunciati ontologici di ciascun provider siano compresi correttamente (per esempio intendendo treno come oggetto e non come evento) assicurando di conseguenza la validità delle mappature tra schemi?

La semantica FOL non è dunque in grado di risolvere i problemi che nascono in infrastrutture SOA di una certa complessità. In breve, per poter gestire la conformità semantica in un contesto FOL, tutte le implementazioni (sia dei provider che dei consumatori) dovrebbero essere riunite in una sola, cosa che chiaramente non è possibile nelle infrastrutture service-oriented prive di elementi centralizzati, come SPCoop. Per superare le limitazioni della semantica FOL, le descrizioni dei servizi dovrebbero essere interpretate rispetto alle condizioni che si verificano nell'insieme dei possibili stati dei provider. Questa è la situazione che andiamo ora a esaminare.

2.1.6 La semantica modale

Il logico tedesco Gottlob Frege chiamava Sinn (senso) in contrapposizione a Bedeutung (significato) la descrizione dell'intensione dei predicati logici anziché la loro estensione.⁽⁹⁾ Informalmente, chi sviluppa software può cogliere facilmente la differenza tra intensione ed estensione considerando la differenza tra una query SQL (intensione) e il set di risultati restituito da quella query su una particolare istanza di database in un preciso momento (estensione). Dopo Frege la semantica intensionale è stata studiata come estensione di una logica FOL chiamata logica modale,⁽¹⁰⁾ che introduce la nozione di mondi possibili come insieme di modelli estensionali accessibili all'agente che compie il ragionamento, permettendo così una descrizione formale dell'interpretazione delle formule logiche rispetto a un insieme di situazioni differenti.

Se ora consideriamo un provider come un sistema al quale un client può attribuire un set di possibili stati, allora per ciascun client possiamo sostituire la singola interpretazione FOL con un set di interpretazioni, una per ogni possibile stato del provider. Cosa cambia realmente in questo contesto? La ragione per la quale logici e filosofi hanno adottato la logica modale nel corso dell'ultimo secolo è la possibilità di formalizzare atteggiamenti proposizionali in grado di modellare le diverse maniere in cui un agente può considerare enunciati formali (proposizioni) rispetto al set di mondi possibili al quale ha accesso. Per esempio, data una formula 'f' della descrizione di un servizio (informalmente, il treno Peloritano parte da Palermo), un agente può credere, sapere o ritenere possibile 'f' rispetto al set di stati al quale può accedere, laddove, al contrario, la logica FOL è adatta a descrivere la verità in una situazione singola, fissa e globalmente accessibile. Tali atteggiamenti proposizionali sono formalizzati mediante operatori specifici di secondo ordine associati alle regole che legano l'interpretazione delle dichiarazioni modali al contesto FOL sottostante. Per esempio, nella logica epistemica standard, affinché un agente A possa conoscere una proposizione occorre che quest'ultima sia vera in tutti i mondi accessibili da A; mentre per credere in una proposizione questa condizione non è necessaria.

La logica modale è utile per caratterizzare la semantica delle infrastrutture service-oriented e quindi per migliorare l'interoperabilità? Certamente, dal momento in cui in un mondo aperto di servizi indipendenti e altamente dinamici non esiste un'unica situazione globalmente accessibile, inviare una richiesta a un servizio impone al client di assumere un atteggiamento rispetto alle proposizioni fornite dal provider. Anche se la modellazione dei sistemi distribuiti basata sulla logica modale è ben sviluppata in letteratura,⁽¹¹⁾ la ricerca sull'applicazione di questo paradigma nel campo dell'integrazione dei servizi all'interno di un ambiente Web è relativamente nuova.⁽¹²⁾ Nondimeno, i client di servizi distribuiti all'interno di scenari di interoperabilità complessi assumono di fatto una modalità, e ciò in pratica consiste nel definire policy per il trasferimento dei dati dal provider all'implementazione client che prendano in considerazione in qualche modo il fatto che questi dati riflettono solamente una delle varie situazioni possibili. Sforzi per far emergere questi aspetti modali dall'opacità delle implementazioni adottando modelli trasparenti sono attualmente in corso. La

sintassi legata a recenti proposte di condivisione delle ontologie e scambio di metadati è generalmente capace di rappresentare costrutti di ordine superiore come le metaclassi; ciò permetterebbe agli operatori modali di essere sintatticamente rappresentati. In generale, tuttavia, linguaggi logici riccamente descrittivi come OWL(13) sono concepiti per un singolo agente che accede a un'unica (seppure aperta) situazione. Le metaproprietà come la fiducia(trust) in un servizio sono normalmente lasciate alle caratterizzazioni dei metalivelli unite con una qualche sintassi ai modelli business del primo ordine.(14) E per quanto concerne la semantica formale, recenti studi sulla formalizzazione delle proprietà non funzionali dei Web service puntano a catturare qualche aspetto modale nel senso qui delineato, inclusi temporalità e fiducia.(15)

Si noti che, in un framework modale, la comprensione da parte del client del significato assegnato dal provider ai predicati della descrizione del servizio (ovvero gli elementi XML) può essere considerata come un'atteggiamento proposizionale, in particolare come la credenza nell'efficacia di una certa regola di corrispondenza semantica. Per esempio, un client di servizi forniti dalle Ferrovie Italiane potrebbe credere che un Treno sia inteso dal provider come una sottoclasse di Artefatto, ovvero un Oggetto Fisico, tale che se un Treno è presente nel momento 't' allora ogni sua parte è parimenti presente nel momento 't'.(16) Ciò non è garantito sempre e comunque per una qualche ragione trascendente; è solo la convinzione specifica di uno specifico client che usufruisce dei servizi delle Ferrovie Italiane. In effetti le Ferrovie Italiane avrebbero potuto usare Treno per indicare la corsa di un treno, cioè come una sottoclasse di Evento, che se avviene nel momento t non è necessariamente presente in modo completo nel momento t.(17) Come fanno i client a essere ragionevolmente sicuri che le loro convinzioni sui provider siano fondate? Certamente, disporre di servizi descritti in un linguaggio dati nel quale possano essere espressi i vincoli basilari del primo ordine risulta assai utile: come distinguere Treno come Oggetto da Treno come Evento senza un mezzo per esprimere le dipendenze di inclusione (cioè le relazioni è-un) della forma $Treno(x) \rightarrow Oggetto(x)$? In effetti, più vincoli logici possono essere espressi in modo standard, più client possono verificare la validità dei modelli del provider e caricarli nella propria base di conoscenza con una certa garanzia formale. I vincoli formali possono inoltre assistere la comprensione semantica dal momento che predicati non chiari possono essere interpretati alla luce di un numero di predicati basilari maggiore rispetto a quelli a cui possono essere legati. I vincoli logici, tuttavia, non sono sufficienti: una buona logica sarebbe inutilizzabile se predicati e assiomi venissero scelti erroneamente. Chi può affermare per esempio che la distinzione tra Oggetti ed Eventi è cruciale dal momento che è la radice di due modi differenti di essere nel tempo e che è essenziale inserire questa distinzione al vertice di una gerarchia delle dipendenze? Dove viene mantenuto un set omogeneo di assiomi di questo tipo di significati basilari, e come, nel caso, possono essere condivisi questi fondamenti semantici? il momento di parlare di ontologie.

2.1.7 Il ruolo delle ontologie

Nella linea di Quine, consideriamo un'ontologia come una descrizione intensionale di quanto esiste in un determinato dominio, dove esistere ha il senso di valore di una variabile vincolata.(18) Pertanto, in un framework come quello prima descritto, qualunque descrizione di un servizio corrisponde nei fatti a una ontologia. I linguaggi di descrizione dei servizi come WSDL permettono di utilizzare l'espressività degli schemi XML per descrivere tipologie di dati atomiche e composte che possono essere considerate come strutture di vincoli e predicati logici, che a loro volta permettono di specificare una sorta di descrizione intensionale. Tuttavia qualunque provider è autorizzato a creare la propria ontologia e non vi è ragione di presupporre omogeneità o alcuna euristica unificante immanente come, per esempio, la similitudine dei simboli dei predicati (le tag XML) basata sulle similitudini dei linguaggi naturali. In particolare, SPCoop non pone attualmente vincoli rispetto alle ontologie da usare per la qualificazione semantica dei servizi e dei dati negli Accordi di Servizio. Interpretare l'ontologia di qualcun altro vuol dire esprimere congetture, a meno che un insieme standard di predicati non venga adottato, condiviso, compreso e uniformemente interpretato da ogni nodo dell'ambiente distribuito. La disponibilità e l'accettazione di standard semantici è una condizione abilitante per la diffusa adozione di infrastrutture service-oriented.(19) Per questa ragione numerose persone sia nel settore della ricerca che nell'industria sono al lavoro su questi aspetti. Numerosi standard di contenuto che puntano ad astrarre la comune concettualizzazione di un settore verticale in dizionari di dati ben definiti sono già disponibili per gli scambi elettronici in specifiche comunità. Dizionari di dati business standard sono spesso usati come dorsali semantiche per implementare infrastrutture di integrazione servizi di tipo hub-and-spoke.(20)

Non è obiettivo di questo documento fornire indicazioni sull'adozione e lo sviluppo di ontologie nel contesto di SPCoop: un argomento complesso e fondamentale come questo richiederebbe un attento approfondimento. Tuttavia, la diffusione di standard per il Web semantico come RDF (Resource Description Framework) e OWL, nonché lo standard di annotazione SAWSDL, ha posto le basi per questo sviluppo, e le applicazioni di schemi concettuali orientati alla semantica nei Web service sono da tempo oggetto di studio e sperimentazione.(21) Non tratteremo qui dei risultati di queste ricerche, ma è importante sottolineare come l'adozione di questi linguaggi - anche se per mezzo di implementazioni parziali - agevola lo sviluppo e la manutenzione di ontologie estese in modo distribuito. Per esempio, definire le dipendenze di inclusione in termini di condizioni di appartenenza sufficienti e necessarie come consente di fare OWL alleggerirebbe l'onere di sviluppare concettualizzazioni sotto forma di strutture complete di predicati primitivi aumentando così la copertura della semantica di business. Per esempio, posto che:

$$\text{ServizioAutoAlSeguito}(x) \rightarrow \text{Servizio}(x) \quad (22)$$

un concetto come il seguente:

$$\text{TrenoConServizi}(x) \wedge \text{Treno}(x) \rightarrow \exists \text{Servizio}(y) \rightarrow \text{possiedeServizio}(x, y)$$

permetterebbe di dedurre la seguente inclusione:

$Treno(x) \rightarrow ServizioAutoAlSeguito(y) \rightarrow possiedeServizio(x, y) \rightarrow TenoConServizi(x)$
eliminando quindi dall'ontologia la seguente primitiva:

$TrenoConServizioAutoAlSeguito(x) \rightarrow TenoConServizi(x)$

Adottare una solida concettualizzazione top-level, ovvero un insieme ristretto di concetti basilari validi per qualunque dominio (per esempio SUMO(23) o DOLCE(24)), riduce il rischio di inserire vincoli e predicati non chiari difficilmente interpretabili dalla maggior parte dei client. Per esempio, l'introduzione di un concetto come questo:

$VagoneDieciMinuti(x) \wedge Vagone(x) \rightarrow Durata(x, 10)$

potrebbe essere evitata se Vagone e Durata fossero legati a una ontologia top-level nella quale:

$Durata(x, y) \rightarrow Evento(x)$ (la durata temporale è definita per gli eventi)

$Oggetto(x) \rightarrow \neg Evento(x)$ (gli oggetti non sono eventi)

$Vagone(x) \rightarrow Oggetto(x)$ (i vagoni sono oggetti)

dal momento che la maggior parte dei ragionamenti automatici (25) sarebbero in grado di determinare che in ogni modello:

$\{x | \exists (y) Vagone(x) \rightarrow Durata(x, y)\}$ (non esistono vagoni temporali)

In conclusione, come hanno evidenziato gli ontologi, la ragione per investire nel livello ontologico è che condividendo una concettualizzazione corretta si riducono le potenziali incomprensioni e difformità causate dal conflitto tra modelli.(26) D'altra parte, creare ontologie corrette e complete è alquanto difficoltoso, e acquisire, valutare e adottare le risorse ontologiche disponibili non è compito facile. Il relativismo ontologico, per di più, impedisce una naturale convergenza globale e trasparente di un set limitato di ontologie universalmente accettate e suggerisce la possibilità che, in numerosi domini di business, si dovrà probabilmente continuare a lavorare con più modelli concettuali eterogenei eppure coesistenti.

2.1.8 Modelli di interoperabilità semantica

Le tecnologie e gli standard SOA forniscono ai progettisti una serie di soluzioni per l'interoperabilità semantica nel senso descritto in precedenza. Nei paragrafi che seguono illustreremo le principali caratteristiche di alcuni possibili approcci all'interoperabilità semantica. La scelta di un particolare approccio dipende dall'ambiente di business nel quale viene installata l'infrastruttura a servizi; questi possibili approcci sono basati su criteri di classificazione che catturano gli elementi essenziali dal punto di vista dell'integrazione semantica affrontando le problematiche principali della gestione della semantica all'interno di sistemi service-oriented distribuiti.

I differenti modelli di interoperabilità semantica sono classificati in base a due dimensioni fondamentali: 1) la scelta tra due possibili modalità di definizione delle mappature di integrazione, la prima nella quale lo schema di ciascun servizio è mappato rispetto a quello di ogni altro (any-to-any), e la seconda nella quale ciascuno schema è mappato rispetto a un unico altro schema (any-to-one); 2) la scelta tra un'esecuzione della logica di integrazione in un unico nodo

distinto (centralizzata) oppure un'esecuzione distribuita tra più nodi funzionalmente equivalenti (decentralizzata). Combinando questi criteri di classificazione otteniamo quattro modelli di interoperabilità.

2.1.9 Modello any-to-any centralizzato

Nel modello any-to-any centralizzato illustrato in Figura 3, i servizi sono interpretati e mappati gli uni rispetto agli altri senza ricorrere a ontologie. La funzione di integrazione semantica è fornita da un componente speciale, l'integratore, che crea un modello per ciascun servizio partecipante sulla base della propria interpretazione della descrizione del servizio. La figura mostra come l'integratore mantenga un modello per ciascun servizio (ellissi interne) in base all'interpretazione della rispettiva descrizione del servizio. Dall'altra parte, i provider mantengono i propri modelli realizzati attraverso le loro implementazioni. Le mappature, tuttavia, vengono applicate direttamente tra coppie di modelli interni. Chi progetta queste applicazioni determina dunque la logica di integrazione mappando liberamente i dati in ingresso e in uscita da qualunque servizio del sistema a qualunque altro secondo la propria comprensione del significato effettivo delle descrizioni rese disponibili dai provider dei servizi. Il modello any-to-any centralizzato viene impiegato nella definizione dei processi di business con il supporto di linguaggi standardizzati come BPEL,⁽²⁷⁾ utilizzati in ambienti di produzione maturi. Analizziamo ora qualche implicazione pratica di un modello di integrazione di questo genere.

[Figura 3]

Negli scenari any-to-any centralizzati i provider di servizi sono tipicamente neutrali rispetto ai problemi di integrazione; il loro contributo all'infrastruttura è limitato alle interfacce del servizio e non assumono alcun impegno né accettano alcun vincolo per integrare i rispettivi servizi in qualunque applicazione sovrastante. I provider non vengono coinvolti dai progettisti ai fini di rendere semanticamente omogenea l'integrazione, pertanto gli integratori possono liberamente adoperare le interfacce del servizio senza bisogno di alcuna negoziazione preventiva o successiva con i provider. I servizi sono normalmente atomici, indipendenti e autocontenuti; non sono influenzati da quanto accade esternamente al loro ambiente. Qui risiede contemporaneamente la forza e il limite di questo approccio: anche se i provider fossero disponibili a cooperare all'opera di integrazione, non potrebbero tuttavia farlo a causa della necessità di fornire le specifiche complete delle proprie mappature rispetto a qualunque servizio che i progettisti intendessero includere. In pratica, per poter contribuire al processo di integrazione, ciascun provider dovrebbe ricevere il piano di integrazione, analizzarlo e specificare le mappature dei propri linguaggi dati rispetto a quelli degli altri servizi, cosa che richiederebbe accordi bilaterali tra coppie di partecipanti - un processo davvero disagiata.

In effetti, nelle applicazioni di coreografia - quelle incaricate di comporre e coreografare servizi allineati al business - i provider non possono dare contributi semantici di alcun genere: i progettisti di coreografie sono le uniche autorità semantiche e il comportamento complessivo del sistema è il prodotto

della loro comprensione (che sono in realtà credenze) dei servizi in base alle descrizioni disponibili. In questo contesto l'erronea comprensione dell'attività di un servizio può causare errori nell'ambito dello scambio dei dati propagabili all'intera applicazione coreografica. Inoltre, a differenza delle librerie utilizzate nella normale programmazione software, questo genere di errore può portare al malfunzionamento di sistemi forniti da terze parti. Poiché gli errori semantici possono essere rilevati solamente in fase di runtime e configurare ambienti di test per le infrastrutture di servizi non è questione da poco, esiste la concreta possibilità che tali errori possano verificarsi in qualunque momento durante la vita del sistema con conseguenze imprevedibili. dunque essenziale ridurre il più possibile il rischio di incomprensioni semantiche. Come abbiamo visto, le entità con cui ha a che fare un provider di servizi non sono affatto limitate ma possono comprendere eventi, proprietà, astrazioni e qualunque altra cosa i progettisti abbiano ritenuto appartenere al dominio del sistema. Una tale decisione (ovvero gli enunciati ontologici) non è data per natura né causata da qualcosa inerente il business di per sé; piuttosto, è una scelta di design. A questo punto o chi progetta coreografie è in grado di presupporre con certezza che i servizi che intende integrare condividano il medesimo set di enunciati ontologici, oppure occorre trovare un modo per acquisire il significato effettivo delle descrizioni dei servizi e impostare una trama complessa di mappature concettuali.

All'interno di ambienti chiusi, come quelli delle grandi aziende, i provider di servizi possono essere obbligati a seguire la medesima ontologia. Ovviamente ciò risolve tutti i problemi in anticipo, ed è infatti la miglior opzione perseguibile quando disponibile. In un caso come questo il team che supporta l'infrastruttura di servizi verrebbe in pratica coinvolto in un processo di standardizzazione ontologica il cui impatto andrebbe attentamente analizzato e gestito. Integrare database e applicazioni legacy, per esempio, farebbe insorgere il problema non secondario dell'armonizzazione delle concettualizzazioni che questi componenti introducono nell'ontologia aziendale. Quando invece l'adozione di un'ontologia standardizzata rimane fuori portata, è sempre possibile che client e provider appartengano a unità organizzative strettamente legate insieme, e che i responsabili della coreografia abbiano accesso alle implementazioni dei servizi (per esempio potendo consultare documentazione interna, che probabilmente contiene molte più informazioni di quelle fornite dall'interfaccia esposta). Ciò offre la possibilità di controllare l'interpretazione fornita dal provider all'interfaccia pubblica garantendo la corretta interpretazione di operazioni e dati all'interno dell'applicazione coreografica. Questo è lo scenario ideale per sviluppare un'integrazione di servizi mediante modelli any-to-any centralizzati. Sfortunatamente, però, questo non è l'unico scenario che i progettisti normalmente devono affrontare.

Cosa può aiutare i 'coreografi' a interpretare le specifiche di un servizio quando non esistono ontologie standardizzate e le uniche informazioni disponibili si trovano nell'interfaccia pubblica? Ricordiamo che la misura dell'uniformità semantica nell'uso di un servizio è correlata a come l'interpretazione del client corrisponda a quella del provider; un modello presente nel set di mondi possibili di un provider dovrebbe essere un modello anche per il client. Alla luce di quanto abbiamo discusso sinora, l'unica euristica disponibile proviene dal linguaggio

descrittivo purché esso sia capace di esprimere vincoli logici (per esempio dipendenze di inclusione). Un linguaggio dotato di vincoli condurrebbe l'interpretazione del client attraverso una griglia di predicati descrittivi così da consentire una specie di 'comprensione olistica' di ciascun singolo predicato in base all'intera ontologia. Purtroppo il linguaggio standard delle SOA, ovvero XML, non standardizza i vincoli di modellazione logica - una grave limitazione quando si sviluppano applicazioni di coreografia in ambienti loosely-coupled. Ancora una volta ciò evidenzia il valore delle estensioni di XML orientate alla semantica come RDFS (RDF Schema) od OWL. Il supporto di questi standard da parte dell'industria è nella fase iniziale dello sviluppo e la loro adozione è ancora limitata, e tuttavia la caratterizzazione semantica dei servizi è uscita dall'ambito accademico ed è un concetto pienamente riconosciuto in ambito industriale.

In conclusione i modelli di integrazione any-to-any centralizzata, come quelli usati nei linguaggi coreografici, sono ideali per ambienti chiusi come le infrastrutture di integrazione enterprise. Adottare un modello di questo genere per l'integrazione di servizi complessi in ambienti aperti richiederebbe ulteriori vincoli e presenterebbe alcuni rischi che devono essere previsti e gestiti.

2.1.10 Modello any-to-one centralizzato

Come mostrato nella Figura 4, nel modello any-to-one centralizzato i dati di input/output sono mappati su un'unica ontologia gestita da un servizio o un'applicazione specializzati (il componente Integratore nella Figura 4). Come nella Figura 3, le ellissi interne indicano che l'integratore dispone di un modello per ciascun servizio, in base alla sua interpretazione delle descrizioni disponibili, mentre i provider derivano i propri modelli per mezzo delle implementazioni. Le mappature, tuttavia, afferiscono alla particolare ontologia mantenuta dall'integratore.

[Figura 4]

L'ontologia (talvolta chiamata modello di business) è un linguaggio dati onnicomprensivo che rappresenta un superset semantico dell'unione di tutti i linguaggi dati esportati dai service provider. Questo è il modello impiegato, per esempio, in una varietà di sistemi content-aware a raggiera come i message broker nelle architetture basate sui messaggi, capaci di tradurre e istradare i messaggi sulla base dei loro contenuti,(28) o il più generico Enterprise Service Bus.(29)

In genere i provider non sono coinvolti nello sviluppo della logica dei nodi integratori; chi progetta un integratore è pertanto responsabile di realizzare il layer semantico. L'uniformità semantica del sistema dipende dalla comprensione e dalle convinzioni dei progettisti, pertanto quanto rilevato nella discussione dei modelli any-to-any centralizzati si applica anche qui. Inoltre i progettisti sono chiamati a mappare le descrizioni del servizio rispetto a un'unica dorsale ontologica immutabile, cosa che richiede competenze ulteriori in confronto al compito di mappare i servizi caso per caso sulla base di conoscenze locali e trasformazioni ad hoc. Da qui, l'adozione di modelli di informazione business quale fondamento semantico unificato per le infrastrutture orientate ai servizi

conduce a un aumento dei costi, almeno in termini di competenze richieste per lo sviluppo della logica di integrazione. Ne vale economicamente la pena?

Si sottolinea spesso come il numero di mappature richieste per l'intero sistema venga significativamente ridotto nei modelli any-to-one passando (in casi limite) da $N \times (N - 1)$ a N , dove N è il numero di servizi coinvolti. Tuttavia la riduzione del numero di mappature non costituisce la differenza più importante: la vera discriminante è invece l'esistenza di un modello business. Infatti, laddove le applicazioni any-to-any polverizzano la semantica in una griglia di mappature a coppie, i modelli any-to-one rendono la semantica un oggetto tangibile e gestibile composto da una ontologia del dominio business e da un set di mappature che traducono le descrizioni fornite dal provider nei termini del modello business.

Nella sezione precedente abbiamo brevemente discusso l'impatto dato dall'adozione di ontologie. Nei modelli di interoperabilità any-to-one la completezza dell'ontologia rappresenta il principale aspetto da risolvere: come possiamo garantire a priori che il modello business adottato contenga ogni cosa abbracciata dalle definizioni del servizio? Se non esistono limitazioni al numero e alla varietà di servizi da integrare ciò generalmente diventa impossibile, tanto che questi sistemi devono prevedere metodi appropriati per estendere l'ontologia al fine di integrare nuove entità di business. L'estensibilità del modello business è quindi essenziale per il successo di questo schema di integrazione. Linguaggi ontologici imperniati sulle 'proprietà', come quelli basati su RDF Schema, permettono di implementare questa caratteristica in maniera adeguata.

L'integrazione di dati all'interno di questo modello può essere implementata in tre modi.(7) Nell'approccio Local-As-View (LAV), le mappature procedono dagli schemi sorgente alla visione globale (cioè l'ontologia) associando singoli elementi dell'alfabeto dati esportati dai servizi alle viste (query) del modello business. L'approccio Global-As-View (GAV), invece, prevede l'esatto contrario: le query sull'alfabeto dati delle fonti vengono mappate sui singoli elementi dell'ontologia. Un terzo approccio, Global-Local-As-View (GLAV), comprende elementi di entrambi i sistemi e costituisce la scelta tipica nei sistemi di data exchange.(30)

Supponiamo per esempio che il provider Ferrovie Italiane esporti il predicato Biglietto(treno, da, a, classe, prezzo) verso un sistema il cui modello business contenga Tratta(ID, inizio, fine) e Tariffa(tratta, mezzo, costo). Si veda la Figura 2, considerando però che il tipo di modello qui presentato potrebbe essere any-to-any o any-to-one a seconda del fatto che il broker utilizzi ontologie o meno. Una mappatura LAV potrebbe essere:

$$Biglietto(f, t, p) \rightarrow \exists(r)(Tratta(r, f, t) \rightarrow Tariffa(r, treno', p))$$

Si noti come in questo esempio alcuni attributi che caratterizzano Biglietto siano assenti dalla traduzione (in particolare quelli che si riferiscono al concetto di treno), mentre il valore ID di Tratta dovrebbe essere inserito nel trasferimento dei dati. Infatti l'approccio LAV è utile se il sistema di integrazione si basa su una ontologia stabile e completa, altrimenti buona parte dell'informazione fornita dalla fonte potrebbe andare perduta. D'altra parte, i servizi hanno a disposizione una certa flessibilità nell'entrare o nel lasciare l'infrastruttura;

un nuovo servizio che entri nel sistema condurrebbe all’arricchimento del set di mappature senza modificare il modello business, sempre che per ciascun elemento del linguaggio dati del servizio possa essere stabilita una adeguata mappatura sullo schema globale.

Nei sistemi di integrazione dati, rispondere a query formulate sullo schema globale utilizzando mappature LAV è notoriamente difficoltoso dal momento che la vista globale è caratterizzata da mappature potenzialmente complesse verso le fonti dati disponibili, e inferire la modalità d’impiego della caratterizzazione al fine di rispondere a query globali non è per nulla semplice, in molti casi. Usare un modello di informazioni business per supportare gli algoritmi di scambio dei dati che traducono i messaggi provenienti da un servizio mappato in messaggi diretti ad altri servizi mappati potrebbe rivelare il medesimo tipo di difficoltà; si consideri per esempio il problema di decidere se istradare alle Ferrovie Italiane, come descritto nella Figura 2, un messaggio di richiesta proveniente da Grand Tour Agency con un contenuto del genere:

$trovaTratta(x, Palermo', Roma') \rightarrow Tariffa(x, trasportoterrestre', y)$

Dal punto di vista di un broker, questa decisione richiederebbe di determinare che treno è un tipo di trasporto terrestre e che di conseguenza le Ferrovie Italiane siano una fonte dati ammissibile. A seconda dell’espressività del linguaggio ontologico, una decisione come questa potrebbe comportare algoritmi complessi, e il sistema di brokeraggio potrebbe essere conseguentemente influenzato da problemi di scalabilità. Ovviamente semplici corrispondenze one-to-one tra proprietà locali e globali potrebbero essere calcolate a costi ragionevoli, ma un linguaggio di mappatura mediocre richiederebbe probabilmente una continua azione di espansione e adattamento del modello business ai linguaggi dati esportati dai servizi, con una forte diminuzione della flessibilità.

L’approccio GAV in un sistema di integrazione di servizi funzionerebbe al contrario: i concetti del modello business verrebbero mappati rispetto alle viste degli schemi sorgente. Facendo riferimento alla Figura 2, presupponendo che lo schema del broker coincida con l’ontologia, una mappatura GAV con lo schema dell’agenzia assumerebbe questo aspetto:

$Itinerario(x) \wedge In(x, a) \wedge In(x, b) \rightarrow \exists(r)(Tratta(r, a, b))$

Gli algoritmi come quelli necessari a calcolare il routing sulla base dei contenuti sarebbero molto più semplici se i sistemi di integrazione fossero limitati a questo genere di mappatura, dato che richiede di specificare l’esatto modo di estrarre (inserire) istanze di concetti globali dalle (nelle) fonti disponibili. Constatiamo comunque come le mappature GAV non siano particolarmente adatte a una perfetta integrazione di fonti dati impreviste in un determinato modello business globale all’interno di ambienti dinamici, come normalmente richiesto nelle infrastrutture di servizi. In questo approccio, infatti, il modello business è prioritario, e le fonti (ovvero i servizi) sono generalmente considerate come un modo per fornire al modello estensioni adeguate. Mescolare risultati GAV e LAV porta al cosiddetto approccio GLAV, nel quale le viste sulle fonti locali possono essere mappate rispetto alle viste sull’ontologia. Questo genere di mappatura è il più efficace e generico ma, ovviamente, impone quantomeno la complessità associata all’approccio LAV.

Riepilogando, i modelli di integrazione any-to-one centralizzati sono efficaci e adatti alla gestione di ambienti complessi e dinamici come i Web service negli scenari business-to-business. Nondimeno, questo approccio richiede uno studio accurato dei compromessi tra espressività dei linguaggi ontologici, generalità delle mappature degli schemi e praticità degli algoritmi di calcolo. Soluzioni any-to-one centralizzate solide dovrebbero essere basate su ontologie di alta qualità e linguaggi di mappatura espressivi, ma la conseguente capacità di effettuare ragionamenti complessi è vincolata ai limiti della scalabilità del sistema di integrazione.

2.1.11 Modello any-to-any decentralizzato

Dalla Figura 5 si evince come i sistemi any-to-any decentralizzati siano quelli i cui servizi gestiscono la propria integrazione con ogni altro servizio senza ricorrere ad alcun modello business o componente di integrazione sovrastante. Per questa ragione possiamo descriverli come sistemi peer-to-peer puri, dove la purezza non dipende solo dal fatto che ciascun nodo della rete è potenzialmente provider e client nello stesso tempo, ma anche dall'assenza di un modello di informazione comune o utilizzo di un servizio di integrazione centralizzato. Notiamo come, sebbene la maggior parte dei sistemi peer-to-peer sia ben lungi dall'essere pura, le infrastrutture di Web service basilari ricadano esattamente in questa categoria dal momento che l'unico servizio comune eventualmente usato dai provider è una semplice repository per la pubblicazione che non coinvolge in alcun modo l'attività di runtime dei servizi stessi. Per inciso, l'attuale disegno architetturale di SPCoop ricade in questa categoria. La Figura 5 mostra tre peer, A, B e C, ciascuno dotato del proprio modello degli altri due (ellissi ricorsive). La logica di integrazione è distribuita tra i peer ed è implementata mappando i modelli l'un l'altro.

[Figura 5]

Nei sistemi peer-to-peer puri, come nelle applicazioni coreografiche standard, l'integrazione consiste in un intreccio di coppie di mappature che potenzialmente legano lo stato di ciascun provider a quello dell'altro. Come nei modelli any-to-any centralizzati, i provider non mantengono alcuna semantica globale, cosa che in genere rende l'approccio peer-to-peer robusto e scalabile. A differenza delle coreografie, però, nei sistemi peer-to-peer le mappature non sono stabilite da un'applicazione centrale; piuttosto, sono distribuite in tutta la rete insieme con gli algoritmi che eseguono la logica di integrazione. In un certo senso, dunque, qualunque servizio può contribuire a una sorta di 'semantica emergente'.(31) Nessuno esercita il controllo dell'intero insieme di mappature; pertanto possono verificarsi mappature cicliche che rendono il sistema di integrazione esposto al rischio di loop non rilevabili qualora questo non sia progettato adeguatamente.(32)

Da un punto di vista teorico, si può mostrare come l'integrazione di dati nei sistemi peer-to-peer puri richieda il ricorso alla logica modale (nel senso spiegato nella sezione Semantica modale) per meglio rispondere a una serie di requisiti come modularità, generalità e trattabilità.(33) In effetti, integrare un

set di provider indipendenti e isolati mediante un approccio del primo ordine richiederebbe un agente supervisore per garantire almeno che gli assiomi da cui dipendono le mappature siano omogenei. Ma la supervisione della logica di integrazione eliminerebbe i vantaggi dell'approccio peer-to-peer poiché imporrebbe a ciascun provider di cooperare con l'infrastruttura a livello globale, e una partecipazione federata così vincolata condurrebbe a una diminuzione della robustezza, della scalabilità e della flessibilità. D'altra parte, un framework di integrazione capace di affrontare il concetto di mondo possibile sarebbe in grado di gestire le difformità, i cicli, le fonti non affidabili e tutti gli altri problemi che possono insorgere in un ambiente incontrollato e auto-organizzante che renda l'integrazione realizzabile dal punto di vista computazionale pur senza imporre un'autorità centrale.

Come abbiamo visto, in un framework intensionale ciascun servizio viene modellato come un agente che contribuisce al set di possibili mondi accessibili da parte dei client. Una mappatura non è interpretata come un'implicazione materiale di formule FOL,⁽³⁴⁾ bensì come trasferimento di informazione dalla fonte verso il bagaglio di conoscenza del client secondo un atteggiamento proposizionale del client nei confronti della fonte. In pratica, un tale atteggiamento è una policy che detta il modo in cui un client raccoglie informazioni dai provider tenendo conto del fatto che essi riflettono un set di modelli coesistenti. Formalizzando le policy di trasferimento dati, un sistema può trattare con qualunque provider di informazioni - sia esso affidabile e perfettamente compreso piuttosto che incerto e oscuro - introducendo regole esplicite inerenti fiducia, preferenze, specifici ragionamenti temporali e così via.⁽³⁵⁾

Le infrastrutture di servizi decentralizzati necessitano inoltre di un modo per risolvere i circoli viziosi che potrebbero essere inseriti dalla distribuzione e proliferazione incontrollate di mappature all'interno del sistema. Ciò richiede che i vari peer adottino una regola che consenta l'identificazione delle transazioni nell'intero sistema per evitare la diffusione di messaggi di richiesta che verrebbero riprocessati anche dagli stessi nodi da cui provengono. Ciò è facilmente implementabile negli ambienti orientati a servizi speciali, come le grid, in cui la transazioni distribuite possono essere identificate uniformemente.⁽³⁶⁾ Problemi tecnici a parte, occorre notare come il ragionamento intensionale aiuti a risolvere la questione dei cicli di mappatura all'interno di un framework teorico solido permettendo di progettare algoritmi di integrazione in modo decentralizzato.⁽³³⁾

Per concludere, l'interoperabilità semantica peer-to-peer spesso non è un'opzione: si tratta dell'unico modello disponibile quando non esiste la possibilità di condividere un'ontologia né ricorrere a servizi di integrazione centralizzati. Sebbene le infrastrutture di servizi siano peer-to-peer per natura, framework completi per gestire questi modelli si trovano ancora in fase sperimentale. In ogni caso, chi progetta infrastrutture di questo tipo deve affrontare un'incertezza inerente dovuta al fatto che la semantica si trova distribuita in sistemi isolati l'uno dall'altro. Alcuni suggerimenti su come modellare questa incertezza con gli strumenti concettuali della logica modale sono stati presentati in questa sezione.

2.1.12 Modello any-to-one decentralizzato

Come illustrato nella Figura 6, i sistemi any-to-one decentralizzati sono infrastrutture peer-to-peer nelle quali gli endpoint si collegano e scambiano informazioni l'uno con l'altro, ma le descrizioni dei servizi sono mappate su (o direttamente prelevate da) un modello business comune condiviso. La Figura 6 mostra come tutte le descrizioni dei servizi appartengano effettivamente a un'ontologia condivisa, e come ciascun servizio modelli gli altri con interpretazioni di frammenti ontologici che corrispondono all'unione dei loro schemi (ellissi ricorsive).

[Figura 6]

Queste ontologie condivise possono essere erogate da specifici servizi di metadati oppure semplicemente fornite sotto forma di vocabolari indirizzabili in rete (per esempio documenti XML Schema); attualmente questo è il metodo più comune per risolvere il layer semantico nelle architetture peer-to-peer basate sul Web, secondo l'opinione della comunità del Semantic Web. Proposte come il Web Services Modeling Framework, per esempio, si concentrano su questo modello, e i suoi autori affermano che [l'uso di ontologie] assicura l'uniformità della rappresentazione testuale del concetto scambiato e consente a tutte le parti coinvolte di interpretare i concetti allo stesso modo.⁽³⁷⁾ Si osservi che, come dice la frase citata, le ontologie 'permettono' ma non 'garantiscono' la medesima interpretazione di un concetto, e questo per le ragioni esposte nelle sezioni precedenti.

Rispetto ai sistemi any-to-one centralizzati come quelli basati sui servizi di brokeraggio, la decentralizzazione richiede che i provider collaborino strettamente con l'infrastruttura nel suo insieme adottando una 'lingua franca' che possa differire significativamente dalle rispettive concettualizzazioni legacy interne - richiedendo pertanto una certa volontà di adattamento. D'altra parte, laddove l'integrazione centralizzata permette a un singolo sviluppatore (o ad alcuni di essi) di definire la semantica complessiva, le uniche autorità semantiche dei modelli di integrazione decentralizzata sono gli stessi provider. Infatti, anche adottando modelli business affermati, il modo in cui ciascun provider interpreta il vocabolario condiviso rimane nascosto dall'implementazione del servizio, ovvero laddove risiede la semantica. La distribuzione delle responsabilità semantiche, elemento distintivo dei sistemi peer-to-peer, è il fattore cruciale in questi scenari e come tale dovrebbe essere attentamente considerato. Osserviamo che mentre le applicazioni centralizzate sono basate su decisioni esplicite di come le descrizioni dei servizi dei provider indipendenti vengono mappate l'una con l'altra - o corrispondano a una qualche ontologia di riferimento - i modelli decentralizzati richiedono invece di intraprendere decisioni semantiche indipendentemente nell'intero ambito della rete. Nondimeno, come regola, queste decisioni non vengono valutate da alcuna autorità globale, e il processo di convergenza verso una semantica uniforme può essere influenzato ma non controllato. Dopo tutto, in un ambiente totalmente decentralizzato senza alcuna autorità stabilita, chi può dire che un service provider abbia male interpretato un predicato descrittivo di una ontologia condivisa?

Per questa ragione l'insorgere di un inevitabile 'relativismo' costituisce il problema principale quando si adottano concettualizzazioni condivise in ambienti decentralizzati. Come delineato nelle sezioni precedenti, tuttavia, disporre di modelli business specificati in un linguaggio ontologico accurato e basati su un layer di base adeguato riduce il rischio di interpretazioni erronee introdotte dagli agenti nel mappare le rispettive concettualizzazioni su tali modelli. Una adeguata copertura delle primitive concettuali rispetto al dominio di business (completezza) che implichi la possibilità di migliorare progressivamente gli schemi concettuali (estensibilità) è un altro fattore essenziale per il successo dell'adozione di ontologie condivise all'interno di infrastrutture orientate ai servizi. Ma esiste sempre il rischio di imprecisioni, incomprensioni, errori, approssimazioni, mancanza di conoscenza o persino intenti malevoli, specialmente negli ambienti loosely-coupled privi di controllo. Da quanto esposto nelle sezioni precedenti abbiamo visto come la modellazione di questi fattori richieda l'adozione di framework capaci di gestire politiche modali.

In conclusione, per quanto l'integrazione any-to-one decentralizzata sia una soluzione percorribile per gestire infrastrutture di servizi peer-to-peer altamente dinamiche con un livello di centralizzazione minimo, essa impone il compito non facile di uniformare tutti i partecipanti rispetto a una concettualizzazione comune. Per implementare questo modello, la disponibilità di ontologie ampie e solide è necessaria ma non sufficiente: è infatti essenziale gestire policy specifiche che consentano di gestire l'incertezza sempre associata alla semantica di dati provenienti da fonti informative esterne.

2.1.13 In conclusione

Sulla strada dell'integrazione tra sistemi informativi interconnessi in reti complesse come SPC, gli sviluppatori si trovano direttamente ad affrontare una delle discipline più antiche: la semantica. Gli ambienti SOA, con i loro linguaggi descrittivi e i potenti algoritmi di ragionamento, forniscono una base solida e standardizzata che agevola la progettazione e l'implementazione di infrastrutture IT con capacità semantiche. Allo stato attuale dell'arte, le infrastrutture orientate ai servizi stanno rapidamente diventando una realtà, e l'adozione di framework specifici per sviluppare il layer semantico è una delle maggiori priorità. Difficoltà di molti generi accompagnano l'adozione di artefatti, tecnologie, metodologie, pratiche e standard orientati alla semantica, e tuttavia l'informatica moderna provvede gli strumenti fondamentali per sviluppare soluzioni confacenti. In definitiva, riteniamo che la mancanza di consapevolezza sulla natura della semantica, sui motivi della sua importanza, e sulle sue possibilità di una sua efficace modellazione rappresenti l'ostacolo più significativo alla sua applicazione nelle infrastrutture orientate ai servizi.

In questo documento abbiamo offerto un'ampia panoramica del problema dell'interoperabilità semantica nelle infrastrutture orientate ai servizi. Abbiamo fornito una definizione specifica di cosa si possa intendere per caratterizzazione semantica dei servizi in ambito sia estensionale che intensionale, e abbiamo spiegato il ruolo delle ontologie. Abbiamo quindi analizzato quattro modelli base

per l'interoperabilità semantica che dipendono dal modo in cui sono tracciate le mappature tra descrizioni del servizio e ontologie, nonché da dove viene valutata la logica di integrazione. La scelta tra questi quattro modelli dipende da vincoli non funzionali come confini organizzativi, disponibilità di standard semantici e così via. Abbiamo offerto qualche consiglio in merito ai punti sui quali i progettisti dovrebbero concentrarsi nell'implementazione del layer semantico nella gamma di modelli di interoperabilità possibili. Nel contesto di SPCoop, che si caratterizza come un modello decentralizzato, una particolare attenzione deve essere posta all'accuratezza con cui ontologie e annotazioni semantiche vengono sviluppate. Il ricorso a modelli di business condivisi e standardizzati si configura come un'opzione in grado di facilitare il pieno sviluppo delle potenzialità di integrazione della Sistema Pubblico per la Cooperazione Applicativa.

2.1.14 Note

1. Un'ontologia è una teoria su ciò che esiste in un dominio applicativo.
2. RosettaNet, <http://www.rosettanet.org>.
3. Vedere ad esempio, S. Colucci, T. Di Noia, E. Di Sciascio, F. M. Donini, and M. Mongiello, Description Logics Approach: Logic Based Approach to Semantic Matching of Web Services, *Journal of Computing and Information Technology* 11, No. 3, 217 223 (2003).
4. T. J. Taylor, *Mutual Misunderstanding: Communicational Scepticism and the Theorizing of Language and Interpretation*, Duke University Press, Durham, N.C. and Routledge, London (1992).
5. W. V. O. Quine, *Ontological Relativity and Other Essays*, Columbia University Press, New York (1969).
6. Vedere Semantic Web site at W3C: <http://www.w3.org/> Uschold, Where are the Semantics in the Semantic Web? *AI Magazine* 24, No. 3, 25 36 (September 2003).
7. M. Lenzerini, Data Integration: A Theoretical Perspective, *Proceedings of the 21st ACM SIGACT-SIGMODSIGART Symposium on Principles of Database Systems, PODS 2002, Madison, Wisconsin (2002)*, 233 246.
8. S'intende che i simboli dell'alfabeto operativo corrispondano ai metodi di accesso e manipolazione del servizio.
9. F. L. G. Frege, *On Sense and Reference*, *Translations from the Philosophical Writings of G. Frege*, Third Edition, P. Geach and M. Black, Editor and Translator, Blackwell, Oxford (1980).
10. See J. Hintikka, *Knowledge and Belief*, Cornell University Press, Ithaca, NY (1962) and S. Kripke, *Semantical Analysis of Modal Logic*, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 9, 67 96 (1963).
11. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi, *Reasoning about Knowledge*, MIT Press, Cambridge, MA (1993).
12. T. Finin and A. Joshi, Agents, Trust, and Information Access on the Semantic Web, *ACM SIGMOD Record*, ACM, New York (December 2002).
13. Web Ontology Language (OWL), <http://www.w3.org/2004/OWL/>.

14. Semantic Web Trust and Security Resource Guide, <http://www.wiwiss.fu-berlin.de/suhl/bizer/SWTSGuide/>.
15. J. O Sullivan, D. Edmond, and A. H. M. ter Hofstede, Formal Description of Nonfunctional Service Properties, Technical Report, Queensland University of Technology (February 9, 2005),
16. Ci riferiamo alla nozione di continuant in P. Simons, *Parts: A Study in Ontology*, Clarendon Press, Oxford (1987).
17. Ancora ci riferiamo alla nozione di relativismo ontologico in W. V. O. Quine, (5).
18. W. V. O. Quine, *Word and Object*, The MIT Press, Cambridge, MA (1960).
19. J. Jacobs and A. Linden, Semantic Web Technologies Take Middleware to Next Level, Research Note T-17 5338, Gartner, Inc. (August 20, 2002).
20. Siebel Business Integration Common Objects, Siebel Systems, Inc., <http://www.siebel.com/>.
21. W3C Semantic Web Services Interest Group, <http://www.w3.org/2002/ws/swsig/>.
22. Se non altrimenti specificato, le variabili si intendono quantificate universalmente.
23. Suggested Upper Merged Ontology (SUMO), IEEE Standard Upper Ontology Working Group, [http:// suo.ieee.org/](http://suo.ieee.org/).
24. Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE), WonderWeb Foundational Ontology Library, <http://wonderweb.semanticweb.org/>.
25. See, for instance, I. Horrocks, FaCT and iFaCT, Proceedings of the International Workshop on Description Logics (DL 99) (1999), pp. 133 135, and V. Haarslev and R. Moeller, Description of the Racer System and Its Applications, Proceedings of the International Workshop in Description Logics (DL2001) (2001), pp. 132 142.
26. N. Guarino, The Ontological Level, in R. Casati, B. Smith and G. White, Editors, *Philosophy and the Cognitive Sciences*, Ho(R)lder-Pichler-Tempsky, Vienna (1994), pp. 443 456.
27. Business Process Execution Language for Web Services, IBM Corporation, [http://www.ibm.com/ developerworks/library/ws-bpel/](http://www.ibm.com/developerworks/library/ws-bpel/).
28. WebSphere Business Integrator Message Broker, IBM Corporation, <http://www.ibm.com/software/integration/wbimessagebroker/>.
29. Patterns: Implementing an SOA Using an Enterprise Service Bus, IBM RedBook SG24-6346, IBM Corporation (2004).
30. R. Fagin, P. G. Kolaitis, and L. Popa, Data Exchange, Getting to the Core, Proceedings of ACM Symposium of Principles of Database Systems, ACM, New York (2003), pp. 90 101.
31. K. Aberer, T. Catarci, P. CudreYen-Mauroux, T. Dillon, S. Grimm, M. Hacid, A. Illarramendi, M. Jarrar, V. Kashyap, M. Mecella, E. Mena, E. J. Neuhold, A. M. Ouksel, T. Risse, M. Scannapieco, F. Saltor, L. de Santis, S. Spaccapietra, S. Staab, R. Studer, and O. De Troyer, Emergent Semantics Systems, Proceedings of the First International Conference on Semantics of a Networked World (CSNW04), Paris, France (2004).

32. A. Halevy, Z. Ives, D. Suciu, and I. Tatarinov, Schema Mediation in Peer Data Management Systems, Proceedings of 19th International Conference on Data Engineering (ICDE) 2003, pp. 505 516.
33. D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati, Logical Foundations of Peer-To-Peer Data Integration, Proceedings of the 23rd ACM SIGACTSIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2004), ACM, New York (2004), pp. 241 251.
34. Formule if-then tali che se la premessa è vera la conseguenza è anch'essa vera.
35. J. Carter and A. A. Ghorbani, Towards a Formalization of Trust, Web Intelligence and Agent Systems 2, No. 3, 167 183, (March 2004).
36. Grid data integration based on epistemic logic is Peer-to-Peer Data Integration on Grids, Proceedings of the International Conference on Semantics of a Networked World: Semantics for Grid Databases (ICSNW 2004), pp. 144 157.
37. D. Fensel and C. Bussler, The Web Service Modeling Framework, <http://informatik.uibk.ac.at/users/c70385/wese/wsmf.bis2002.pdf>, (extended abstract).

2.2 e-Government: Problemi di Interoperabilità Semantica in un Contesto Distribuito

NOTA: questa sezione è ancora da completare.

Basata su materiale contribuito da:
Barbara Re e Flavio Corradini (Universit di Camerino)

Indice

1. Introduzione 3
2. Bisogno di Riorganizzare la PA 4
3. Integrazione delle Informazioni vs Integrazione dei Processi 5
4. Come scegliere strategie di interoperabilità 7
5. Interoperabilità nell'e-Government 9
 - (a) Definizioni..... 9
 - (b) Caratteristiche 9
 - (c) Interoperabilità semantica 1
 - (d) Framework di interoperabilità 12
6. Semantic Web e Semantic Web Services nel dominio di E-Government 13
7. Progetti di e-government sul tema dell'interoperabilità semantica 14
 - (a) SmartGov 14
 - (b) ICTE-PAN 15
 - (c) E-POWER 16
 - (d) Access-eGov..... 17
 - (e) OntoGov 18
 - (f) HOPS 20

(g)	WEBOCRAT	21
(h)	SemanticGov	22
(i)	DIP	22
(j)	BRITE21	24
8.	Applicazioni di Semantic Web e Semantic Web Service	25
(a)	eGOS	25
(b)	Migrating Small Governments' Websites to the Semantic Web . .	25
(c)	Self-adaptive E-Government Service Improvement Using Semantic-Technologies	27
(d)	Transnational Information Sharing and Process Coordination . .	28
(e)	Government Process Classification Scheme	29
(f)	Governance Enterprise Architecture	30
(g)	Gartner Government Performance Framework	31
(h)	WebDGOntologies	31
(i)	Federal Enterprise Architecture Ontology	32
9.	La situazione italiana: cooperazione applicativa -infrastruttura di comunicazione e cooperazione	33
(a)	Metadati soluzioni a supporto della Cooperazione Applicativa SP-Coop	35
10.	Conclusioni	35

Capitolo 3

Linee-guida e componenti per lo sviluppo e l'uso delle ontologie nella PA

3.1 Cosa sono le ontologie informatiche e come si progettano

Basato su materiale contribuito da: Aldo Gangemi
CNR-ISTC e CNIPA

3.1.1 Introduzione

In filosofia, l'ontologia, branca fondamentale della metafisica, è lo studio dell'essere in quanto tale, nonché delle sue categorie fondamentali. Una particolare ontologia è di solito vista come l'insieme delle entità che quelle categorie vanno a descrivere.

Nell'ambito della scienza dell'informazione e in particolare nella cosiddetta "ingegneria ontologica", il termine ontologia non denota un campo di studio, né un dominio di entità da descrivere, ma una classe di oggetti formali, la cui funzione principale è fornire una "specificazione parziale di un insieme di concettualizzazioni" per un sistema informativo.

Le ontologie informatiche si sganciano inoltre dalla prospettiva abitualmente realista dell'ontologia filosofica, perché sono usate per modellare una realtà qualsiasi (passata, attuale, futura, possibile, controfattuale, di finzione, etc.). In cambio, il loro impiego è di solito previsto in maniera dipendente da uno scopo determinato e socialmente rilevante.

L'unico campo di parziale sovrapposizione fra l'ontologia filosofica e quella computazionale risiede nelle cosiddette *ontologie fondazionali*, che tentano di fornire modelli indipendenti dal dominio e da scopi particolari. Il program-

ma delle ontologie fondazionali in informatica è oggetto di critica e acquista senso soprattutto all'interno di metodologie che mirino al riuso di componenti e in particolare alla creazione di modelli di progettazione (*design patterns*), analogamente a quanto accade nell'ingegneria del software.

Infine, le ontologie informatiche hanno una dipendenza di fatto da linguaggi logici che permettano un compromesso soddisfacente fra *espressività* (cioè la varietà di costrutti logici e la loro capacità di rappresentare le nostre concettualizzazioni) e *complessità* (cioè la difficoltà di ragionare automaticamente su quei costrutti).

Un'ontologia nel senso informatico è quindi un artefatto, con una struttura complessa e possibili funzioni. In quanto artefatti, le ontologie hanno un ciclo di vita, con risorse, agenti, procedure di produzione, soluzioni riusabili, modi d'uso, manutenzione, etc.

Le ontologie sono artefatti costituiti da oggetti informativi, cioè da elementi di un sistema semiotico, ma le cui regole sono di solito formulate in maniera completa e non creativa. L'esempio più comune di ontologie è dato da modelli logici dotati di una *semantica formale*, detti *teorie assiomatiche*. In una teoria assiomatica, un vocabolario di termini o predicati ("costanti predicative") *induce un ordine* in un insieme di individui in un dominio ("costanti individuali"). La semantica formale è fornita dalla *teoria dei modelli*, basata a sua volta sulla teoria degli insiemi. In pratica, l'interpretazione dei predicati è fornita rispetto a sottinsiemi del dominio degli individui. Questa interpretazione si chiama *estensionale*. Per esempio, un predicato "Nazione" avrà probabilmente un'interpretazione estensionale rispetto all'insieme degli individui che il progettista dell'ontologia è disposto a classificare come nazioni, per esempio l'Italia e le altre nazioni riconosciute dall'ONU. Il predicato Nazione sarà poi probabilmente considerato *disgiunto* da altri, come "Persona Fisica", perché nessuna Nazione è anche una persona fisica, almeno nell'intuizione comune.

Una teoria assiomatica contiene anche *assiomi*: gli assiomi sono assunzioni concettuali, spesso implicite nell'uso quotidiano del linguaggio, che permettono di caratterizzare meglio l'interpretazione estensionale dei predicati. Per esempio, il progettista dell'ontologia delle nazioni potrebbe voler caratterizzare le singole nazioni in base al continente di cui il loro territorio è parte, o in base alla loro forma di governo. Utilizzando altri predicati come "Continente", "Territorio", "FormaDiGoverno", "parteDi", "haTerritorio", etc., sarà possibile aggiungere assiomi che caratterizzino il predicato "Nazione", per esempio *assumendo* che se qualcosa è una nazione, allora avrà un territorio che è parte di un continente, e avrà una forma di governo. Come conseguenza, si potrà *asserire* che l'Italia ha un territorio delimitato da confini geologici e da coordinate geografiche: Alpi, Mare di Sicilia, etc., che il suo territorio è parte dell'Europa e la sua forma di governo è la democrazia parlamentare.

Ulteriori semantiche formali per le teorie assiomatiche forniscono interpretazioni *intensionali*. Tralasciemo di indagare qui queste forme interpretative, che non vertono sull'ordine indotto dai predicati sul dominio degli individui, bensì su altri ordini indotti sulle interpretazioni estensionali ("mondi possibili") da criteri esterni, per esempio ordinamenti temporali (ciò che possiamo assumere a

un certo tempo), epistemologici (ciò che crediamo di poter assumere), deontici (ciò che dobbiamo assumere), etc.

In questo modo, abbiamo appena introdotto gli elementi di base di un'ontologia vista come una teoria assiomatica: predicati, assiomi, individui e asserzioni. Questi elementi possono essere codificati usando una lingua naturale, per esempio l'italiano, come abbiamo appena fatto, ma per gli usi informatici occorre un sistema semiotico le cui regole siano finite e non creative e la cui sintassi sia leggibile da un algoritmo in modo deterministico e in un tempo finito.

Figura 3.1 illustra due esempi che usano i due linguaggi più usati per la codifica delle ontologie: la *logica del primo ordine* e la *logica descrittiva* [Baader et al., 2003].

La logica del primo ordine permette di esprimere molti tipi di assiomi e asserzioni di qualsiasi *arità* (cioè relazioni fra due o più individui qualsiasi), mentre le logiche descrittive permettono di esprimere solo pochi tipi di assiomi e solamente asserzioni binarie (cioè relazioni fra due individui). Le logiche descrittive hanno “proprietà computazionali” ben note, per cui è possibile prevedere quanto tempo occorra a una macchina per analizzare un'ontologia scritta in logica descrittiva, verificarne la consistenza (cioè l'assenza di contraddizioni logiche) ed effettuare alcune inferenze tipiche del ragionamento sillogistico, per esempio la *classificazione automatica di istanze*, l'*ereditarietà delle proprietà*, la *sussunzione automatica di classi*, e la *materializzazione di fatti* (vedi scheda 2). Per queste ragioni, alcune logiche descrittive sono attualmente usate come sistemi formali per il linguaggio standard per esprimere ontologie sul *web*, cioè OWL (Web Ontology Language, <http://www.w3.org/2004/OWL/>).

I ragionatori automatici su modelli scritti in sistemi di logica descrittiva permettono di inferire informazioni deducibili dalla forma dei modelli e dalle regole logiche del sistema. I quattro tipi di inferenza automatica prodotte dai ragionatori attuali sono:

Classificazione automatica di istanze. Per esempio, se sappiamo che una Nazione Europea è una Nazione il cui territorio è parte dell'Europa e sappiamo anche che l'Italia è una Nazione ed è parte dell'Europa, il ragionatore inferirà automaticamente che l'Italia è una Nazione Europea.

Ereditarietà delle proprietà. Per esempio, se sappiamo che ogni una Nazione ha un territorio (formula 1 sopra) e che Nazione Europea è una sottoclasse di Nazione, il ragionatore inferirà automaticamente che ogni Nazione Europea ha un territorio.

Sussunzione automatica di classi. Per esempio, se sappiamo che una Nazione Europea è una Nazione il cui territorio è (almeno in parte) in Europa, sappiamo che ogni Nazione Balcanica è una Nazione il cui territorio è in parte esteso nei Balcani e infine che i Balcani sono parte dell'Europa, allora il ragionatore inferirà automaticamente che Nazione Balcanica è una sottoclasse di (è sussunta da) Nazione Europea.

Materializzazione di fatti. Per esempio, se sappiamo che il Territorio Italiano è parte dell'Europa e che “ha parte” è la relazione *inversa* di “parte di”, allora il ragionatore inferirà automaticamente che l'Europa ha come parte il Territorio Italiano. Inoltre se sappiamo che la Sicilia (come isola) è parte del Territorio

- (1) $Nazione(x) \rightarrow \exists(y)(haTerritorio(x, y) \wedge Territorio(y))$
(2) $Territorio(x) \rightarrow \exists(y)(parteDi(x, y) \wedge Continente(y))$
 $Nazione(x) \rightarrow \exists(y)$
(3) $(haFormaDiGoverno(x, y) \wedge FormaDiGoverno(y))$
- (4) $Nazione(Italia)$
(5) $Territorio(TerritorioItaliano_{2007})$
(6) $Continente(Europa)$
(7) $FormaDiGoverno(democraziaParlamentare)$
- (8) $haTerritorio(Italia, TerritorioItaliano_{2007})$
(9) $parteDi(TerritorioItaliano_{2007}, Europa)$
(10) $haFormaDiGoverno(Italia, democraziaParlamentare)$
- (11) $Nazione \sqsubseteq \exists haTerritorio. Territorio$
(12) $Territorio \sqsubseteq \exists parteDi. Continente$
(13) $Nazione \sqsubseteq \exists haFormaDiGoverno. FormaDiGoverno$
- (14) $Italia \in Nazione$
(15) $TerritorioItaliano_{2007} \in Territorio$
(16) $Europa \in Continente$
(17) $democraziaParlamentare \in FormaDiGoverno$
- (18) $\langle Italia, TerritorioItaliano_{2007} \rangle \in haTerritorio$
(19) $\langle TerritorioItaliano_{2007}, Europa \rangle \in parteDi$
(20) $\langle Italia, democraziaParlamentare \rangle \in haFormaDiGoverno$

Figura 3.1: Un esempio di ontologia in logica del primo ordine (1)-(10) e in logica descrittiva (11)-(20)

Italiano e che la relazione “parte di” è *transitiva*, allora il ragionatore inferirà automaticamente il fatto che la Sicilia è parte dell’Europa.

Oltre alle formule logiche, un’ontologia informatica contiene (o dovrebbe contenere) molte altre informazioni, di solito rappresentate in forma di *annotazioni*, che non vengono usate dai ragionatori automatici, ma sono molto importanti per progettare, contestualizzare, ordinare, modificare e utilizzare un’ontologia.

Le ontologie informatiche vengono usate soprattutto per *caratterizzare*, *negoziare* e *ragionare* sul significato di termini, assunzioni implicite, regole e altri elementi di rappresentazione, quando si pongono problemi di comunicazione fra agenti, oppure fra agenti e sorgenti di informazione.

Ontologia è usato in informatica anche in sensi vicini, ma informali, equivalenti per esempio a “tassonomia”, “tesauro”, “catalogo”¹. Si veda anche 3.1.4.

In una visione globale, ogni servizio che richieda manipolazione di informazione è assoggettabile a una rappresentazione ontologica. Programmi scientifici globalizzanti come il *Semantic Web* consistono nella creazione di infrastrutture che realizzino tale visione. Programmi di rilevanza sociale come *e-Government*, *e-Policy*, *e-Democracy*, etc. vanno in direzioni simili. Le ontologie, in quanto rappresentazioni esplicite e operative, potrebbero facilitare il risparmio di risorse e la produzione di informazione strategica di dettaglio, insieme a un miglioramento dei rapporti fra tutti gli attori coinvolti nel sistema economico e sociale. Da un punto di vista industriale, l’utilizzo di ontologie è finalizzato alla realizzazione di servizi “semantic”, cioè servizi le cui motivazioni, strutture, funzioni, comunicazione e sviluppo siano (almeno in parte) rappresentati in forma ontologica. Forma leggibile da agenti artificiali, negoziabile da agenti umani, riusabile dalla stessa o da diverse organizzazioni, usata e/o capita dagli utenti secondo i loro profili, interessi e competenze.

Dal punto di vista applicativo, le ontologie sono già usate in alcuni tipi di servizi informativi o di servizi che richiedono l’uso di sistemi informativi, come per esempio la progettazione di basi di dati, l’interrogazione e la fusione di basi di dati eterogenee, il controllo di conformità dei dati, la gestione della conoscenza in organizzazioni e aziende, (in modalità B2B, B2C, etc.²), la ricerca di informazione su basi di dati distribuite, l’estrazione di informazione da testi, l’indicizzazione e catalogazione di documenti, i servizi web e grid per il commercio elettronico, la condivisione di dati scientifici su larga scala (*e-Science*) e l’*e-Government*, i motori di ricerca, i portali, le reti sociali, gli agenti artificiali, etc.

Dal punto di vista del contenuto, la maggior parte dei gruppi di standardizzazione industriale e istituzionale ha istituito o sta istituendo gruppi di lavoro per produrre ontologie di riferimento nel dominio preso in considerazione, per

¹ Gli strumenti informali di organizzazione della conoscenza sembrano però destinati a essere soppiantati da ontologie nel senso più preciso definito sopra, o tutt’al più a essere trasformati in “ontologie leggere”, cioè modelli formali che utilizzano solo pochi tipi di costrutti logici.

² B2B sta per *Business to Business*, cioè comunicazione fra aziende; B2C per *Business to Consumer*, cioè fra azienda a cliente.

esempio: RosettaNet (elettronica), STEP (manifattura), Acord (assicurazioni), Medbiquitous (medicina), Chematch (farmaceutica), the Open Travel Alliance (viaggi), STAR (automobilistica), HL7 (protocolli clinici), UMLS (terminologia medica), Gene Ontology (biologia molecolare), Agrovoc (agricoltura), etc. I gruppi di standardizzazione testimoniano un bisogno diffuso di armonizzare le concettualizzazioni, non solo i linguaggi di programmazione.

Che si consideri plausibile o no la realizzazione di standard per ogni dominio, il tentativo di armonizzare dovrebbe comunque produrre una grande quantità di ontologie riusabili per avviare i cosiddetti *negoziati di significato*: quando un agente: comunità, istituzione, azienda, individuo, si affaccia sul sistema di riferimento concettuale (web semantico, *Grid*, Intranet, sistemi di raccomandazione, sistemi di controllo della conformità e compatibilità, negoziazione semi-automatica, etc.), potrà scegliere se adottare, specificare o proporre un'alternativa alle ontologie esistenti.

Questa potenzialità sembra indicare la strada non solo verso una globalizzazione più spinta, ma anche verso sistemi in cui il ruolo dell'utente è più determinante. L'effettiva interoperabilità fra servizi semantici dipenderà infatti dalla disponibilità degli utenti ad adottare un'ontologia che li rappresenti fedelmente nel contesto di un certo servizio, almeno finché gli utenti di un servizio avranno voglia di esplicitare i propri diritti e punti di vista.

Inoltre, la riuscita di un'ontologia dipenderà in parte dall'accordo fra fornitori di servizi simili o in catena B2B. Questo potrà creare le condizioni per "cartelli" virtuosi o viziosi, ma in ogni caso sarà possibile verificare il contenuto di tali accordi in base ai vincoli formali e al senso inteso delle costanti formalizzate dalle organizzazioni presenti in una catena/cartello.

In generale, lo spostamento gestaltico che sta avvenendo nell'industria ICT (*Information and Communication Technology*) con l'ingegneria ontologica coinvolge la definizione della *qualità* di un servizio informativo attraverso gli assi della chiarezza concettuale, della negoziabilità semantica, della contestualizzazione e della personalizzazione. In altre parole, più volte riprese nello sviluppo di questo rapporto, si intende passare dal problema dell'interoperabilità di sistema a quella *semantica*. Gli osservatori tecnologici di grandi aziende e studi di consulenza industriale stanno raccomandando l'ingegneria ontologica come una delle più importanti tecnologie informatiche, con una tendenza di crescita molto elevata. Il richiamo alla *razionalizzazione esplicita* della gestione dell'informazione, e con essa delle attività sociali in genere, sembra molto più raggiungibile grazie alle ontologie, nonostante i potenziali usi distorti che se ne potrebbero fare e nonostante la difficoltà di raggiungere un accordo sui contenuti. Anzi è proprio la diversità di vedute e la loro confrontabilità che fornisce prospettive brillanti di miglioramento nella progettazione dei dati e nell'impatto sociale che ne deriva, facilitando l'obiettivo più attraente, cioè l'interoperabilità *organizzativa*.

3.1.2 L'ingegneria ontologica

Un'ontologia informatica, come abbiamo visto, è un sistema formale il cui fine è rappresentare la conoscenza relativa ad un determinato dominio e per un

determinato obiettivo all'interno di uno o più sistemi informativi. Essa è descritta mediante l'utilizzo di linguaggi formali, che consentono di codificare (a) *costanti predicative unarie* (chiamate di solito *classi*, o *tipi*); (b) *costanti individuali* (chiamate di solito *individui* o *istanze*); (c) *costanti predicative relazionali* (chiamate variamente *relazioni*, *proprietà*, *attributi*, *associazioni*, *slot*), usate sia all'interno di *assiomi* fra le classi di appartenenza degli individui, sia direttamente sugli individui; (d) *assiomi* sulle classi (chiamati variamente *vincoli*, *restrizioni*, *regole*), che usano un insieme ristretto di *operatori logici*, fra cui gli operatori booleani, i quantificatori, gli insiemi.

Le ontologie possono essere rappresentate in forme molteplici e utilizzando formalismi di differente complessità ed espressività. Si va da semplici cataloghi e glossari, a più articolate tassonomie e tesauri, fino agli schemi relazionali adottati per descrivere database e alle teorie assiomatiche, che rappresentano le ontologie nel senso proprio trattato qui.

A una maggiore complessità della rappresentazione, e quindi di un maggiore investimento in fase di modellizzazione e sviluppo dell'ontologia stessa, corrisponderà una maggiore precisione logica, che rappresenterà meglio il dominio di interesse, imponendo ad esempio vincoli per l'integrità dei dati, consentendo inferenze etc.³

L'ingegneria ontologica è una costellazione di metodi (talvolta organizzati in metodologie) che servono a *progettare*, *costruire*, *valutare*, *mantenere*, *mappare*, *applicare* e *permettere l'evoluzione* di ontologie. Inoltre l'ingegneria ontologica si occupa di realizzare componenti software in grado di fornire funzionalità e interfacce per la progettazione di ontologie (*editors*), per ragionare su di esse (*reasoners*), per interrogarle, contestualizzarle, aggiornarle, etc. La ricerca in ingegneria ontologica può essere suddivisa (con un certo margine di sovrapposizione) in ricerca sui *linguaggi*, ricerca sui *contenuti*, dei quali si parlerà brevemente nelle sezioni a seguire, e ricerca sui *componenti software*, che è in rapida evoluzione e di cui non parleremo qui.

3.1.3 I linguaggi per la rappresentazione delle ontologie

Un linguaggio formale serve a specificare un contenuto. I linguaggi a cui siamo interessati in ingegneria ontologica sono quelli che permettono la definizione di ontologie e basi di conoscenza, il ragionamento su di esse e la loro interrogazione. Lo sviluppo dei linguaggi formali attualmente utilizzati in ingegneria ontologica è avvenuto attraverso due linee evolutive che si sono incontrate alcuni anni fa: il *web semantico* e la *rappresentazione della conoscenza*.

Il *web semantico* (<http://www.w3.org/2001/sw/>) si è evoluto inizialmente assumendo XML (*eXtensible Markup Language*, <http://www.w3.org/XML/>) e RDF (*Resource Description Framework*, <http://www.w3.org/RDF/>) come linguaggi di base: il primo per la sintassi generale, il secondo per la codifica della conoscenza in forma di *triplette* annidabili entro altre triplette. XML permette la definizione di un qualsiasi linguaggio di annotazione (HTML è uno dei

³La complessità e la formalità di un'ontologia non sono però direttamente proporzionali alla sua sostenibilità e utilità, cf. 3.1.4

linguaggi esprimibili in XML), mediante la creazione di *elementi* appropriati. Gli elementi di XML non sono però *tipizzati* in una maniera che permette la semplice definizione di schemi per la costruzione di ontologie. I cosiddetti *schemi XML* (XSD: *XML Schema Definition*, <http://www.w3.org/XML/Schema>) possono rappresentare ontologie, ma hanno due difetti: lasciano un ampio margine di arbitrarietà al modellatore e non distinguono fra diversi “primitivi” di rappresentazione (classi, relazioni, individui, etc.).

RDF è nato con lo scopo di produrre specifiche di contenuto seguendo una metafora linguistica tipica delle lingue occidentali: la struttura (*tripletta*) “Soggetto-Predicato-Oggetto”, laddove ogni Oggetto può essere a sua volta una tripletta. Questa struttura si è dimostrata più flessibile per la rappresentazione di ontologie e basi di conoscenza, soprattutto dopo l’introduzione dei cosiddetti *schemi RDF* (RDFS), un linguaggio che usa le strutture a tripletta di RDF per dichiarare le classi e le relazioni binarie da usare in basi di conoscenza scritte in RDF. RDFS ha però altri difetti: poiché RDFS manca di una *semantica formale*, è possibile confondere diversi livelli logici nello stesso schema; inoltre, non è possibile rappresentare assiomi sulle classi a parte la sussunzione.

D’altra parte, la cosiddetta *Knowledge Representation* (KR) (<http://www.kr.org>), tradizione di ricerca in intelligenza artificiale, aveva già sviluppato sistemi formali di buona qualità per la codifica della conoscenza, definendoli all’interno di linguaggi di programmazione come LISP e PROLOG. Il vantaggio principale dei linguaggi di KR è la loro *formalità*, cioè sono linguaggi logici con una semantica formale esplicita e spesso con proprietà di complessità computazionale ben note, quindi prevedibili dal punto di vista computazionale, come nel caso delle *logiche descrittive* (<http://dl.kr.org>), come è spiegato nella sezione introduttiva.

Il contatto fra il semantic web e la rappresentazione della conoscenza ha prodotto linguaggi (prima DAML e OIL, attualmente OWL, <http://www.w3.org/2004/OWL/>), dotati di una semantica formale, che si “impilano” sugli strati puramente sintattici di XML e RDF, garantendo che la codifica della conoscenza avvenga secondo criteri formali e di complessità nota.

Un esempio aiuterà a chiarire le differenze fra i linguaggi citati. Immaginiamo di voler rappresentare il fatto che Giovanni lavora nel suo bed and breakfast (b&b) solo la notte e in particolare che l’ha fatto la notte del quattro dicembre duemilaquattro. Avremo bisogno di rappresentare (come minimo): *Giovanni*, *persona*, *lavora*, *b&b*, *notte*, la data quattro dicembre duemilaquattro. In un XSD (<http://www.w3.org/XML/Schema>), quasi tutti i termini da rappresentare (tranne Giovanni) sarebbero molto probabilmente codificati come “elementi” e a questo punto si porrebbe il problema di come rappresentare le relazioni fra loro, che sarebbero elementi anch’esse. Mantenendo le differenze logiche a un livello implicito, lo schema potrebbe rappresentare le proprietà di una persona che lavora solo di notte, ma schemi sviluppati da persone diverse conterrebbero scelte molto diverse e la ricostruzione dell’implicito sarebbe estremamente arbitraria e comunque fuori dalla portata di una macchina.

In RDFS (<http://www.w3.org/RDF/>), una possibile soluzione è quella di rappresentare *lavora in* come una “proprietà” (cioè una costante predicativa binaria), *persona*, *b&b*, *notte* e *data* come “classi” (cioè costanti predicative

unarie), mentre *Giovanni, il suo bÈ e la notte del quattro dicembre duemila-quattro* come “individui”. La proprietà *lavora* avrà come “dominio” *persona* e come “codominio” *bÈ*.

Ma in questo modo avremmo alcuni problemi: primo, intuitivamente “lavora in” dovrebbe avere come codominio *azienda*, non *bÈ*; secondo, come esprimere che una persona lavora in un’azienda a un certo tempo? Una proprietà è una relazione binaria, quindi il tempo non può essere espresso. Terzo, come rappresentare le date in modo da considerare solo le notti?

In OWL (<http://www.w3.org/2004/OWL/>), i primitivi di rappresentazione hanno una semantica formale. Useremo la specie di OWL più espressiva ma ancora nei limiti della complessità computazionale accettata comunemente, cioè OWL(DL). L’esempio potrebbe essere rappresentato come segue⁴.

Introduciamo innanzi tutto le proprietà che ci serviranno. Poiché anche in OWL(DL), come in RDFS, le relazioni possono essere solo binarie, una tipica soluzione (un *design pattern*) consiste nel “reificare” le relazioni non binarie. Le reificazioni più frequenti indicano in realtà “eventi”, quindi una relazione binaria (*object property* in OWL) di “partecipazione” fra oggetti ed eventi e la sua relazione inversa (un altro tipico *design pattern*, stavolta di contenuto, vedi sezione 3.1.4) serviranno al nostro scopo:

```
ObjectProperty(partecipaA
inverse(haPartecipante)
domain(Oggetto)
range(Evento))
```

```
ObjectProperty(haPartecipante
inverse(Partecipa)
domain(Evento)
range(Oggetto))
```

A questo punto occorre anche introdurre una relazione per parlare di periodi della giornata e in particolare della notte come uno di quei periodi:

```
ObjectProperty(haPeriodo
domain(Evento)
range(ParteDelGiorno))
Individual(notte
type(ParteDelGiorno))
```

Occorre anche una relazione che colleghi gli eventi ai tipi di dati temporali disponibili, perché i dati “simbolici”: numeri, stringhe, etc. che un componente software può manipolare direttamente⁵ sono espressi direttamente in XSD.

⁴ Utilizziamo qui la cosiddetta “sintassi astratta” di OWL-DL, molto più leggibile della sintassi basata su RDF

⁵ Invece i dati relativi a persone, eventi, etc. “rimandano” a entità che non sono manipolabili direttamente da un componente software.

Queste relazioni con dati simbolici si chiamano *datatype properties* in OWL:

```
DatatypeProperty(hasData
domain(Evento)
range(xsd:Date))
```

Ora si introducono le classi, usando un operatore logico (“partial”), corrispondente alla semantica dell’*implicazione logica*, per affermare per esempio che i B&B sono un tipo di azienda e un turno di notte è un tipo di evento:

```
Class(Oggetto)
```

```
Class(Evento)
```

```
Class(Persona
partial Oggetto)
```

```
Class(Azienda
partial Oggetto)
```

```
Class(B&B
partial Azienda)
```

```
Class(TurnoDiNotte
partial Evento)
```

```
Class(ParteDelGiorno)
```

La nostra ontologia può ritenersi conclusa con la definizione della classe dei “custodi notturni di b&b”, cioè la classe di tutte le persone che lavorano nel turno di notte in un b&b. In questa definizione compaiono altri operatori logici: *complete*, corrispondente alla semantica dell’*equivalenza logica*, *restriction*, corrispondente a una *astrazione lambda*,⁶ *intersectionOf*, corrispondente alla *congiunzione logica*, *someValuesFrom*, corrispondente al *quantificatore esistenziale*, *cardinality*, corrispondente a un *quantificatore ristretto*, *oneOf*, corrispondente a una scelta su un insieme di valori predeterminati:

```
Class(CustodeNotturmoDiB&B
complete
  Persona
  restriction(partecipaA someValuesFrom(intersectionOf
```

⁶ Un’astrazione lambda è un’espressione logica la cui semantica è equivalente a quella di una classe implicita, cioè non espressa da una costante predicativa, bensì da altri costrutti logici che collettivamente equivalgono alla semantica di quella classe. Nei linguaggi naturali, un’astrazione lambda corrisponde a una *parafraasi*.

```

TurnoDiNotte
restriction(haPartecipante someValuesFrom(B&B))
restriction(haData cardinality=1(xsd:Date))
restriction(haPeriodo oneOf(notte)))

```

Letteralmente, la definizione dei custodi notturni di B&B si interpreta come segue: un custode notturno di b&b equivale (“complete”) a una persona che (“restriction”) partecipa ad almeno un (“someValuesFrom”) turno di notte, a cui (“intersectionOf”) partecipa almeno un (“someValuesFrom”) b&b, a una certa data specifica (“cardinality=1”), esclusivamente (“oneOf”) di notte.

Finalmente si può costruire la base di conoscenza: date, turni di notte, b&b e persone specifiche, istanziazioni di individui (“type”) e di asserzioni (“value”):

```

Individual(04122004
type(xsd:Date))

```

```

Individual(turnoDiNotte04122004
type(TurnoDiNotte)
value(haPartecipante B&BGiovanni)
value(haPartecipante Giovanni)
value(haData 04122004)
value(haPeriodo notte))

```

```

Individual(Giovanni
type(CustodeNottturnoDiB&B)
value(partecipaA B&BGiovanni))

```

```

Individual(B&BGiovanni
type(B&B)
value(partecipaA B&BGiovanni))

```

Come si può notare dall’esempio esteso, molti dei costrutti usati sono inespri-
mibili in RDFS, perché richiedono relazioni inverse, operatori booleani (ex. “in-
tersectionOf”), operatori di definizione logica (“complete”), quantificatori logici
negli assiomi (“someValuesFrom”, “cardinality=1”), assiomi logici con astrazio-
ni lambda (“restrictions”). Si noti che la reificazione usata in OWL(DL), se
usata in RDFS, non ottiene gli stessi effetti, per l’impossibilità di dichiarare
assiomi logici sulle classi: nell’esempio, l’evento spuntino è collegato al tempo,
alla persona, all’azienda e alla notte attraverso gli assiomi logici (“restrictions”)
della nuova classe CustodeNottturnoDiB&B. In cambio di questo rigore logico,
si ottiene la possibilità di annotare con una notevole precisione il senso inteso
di un’ontologia solo apparentemente semplice come quella dell’esempio, nonché
la possibilità di fruire di servizi “semantici” come la *classificazione automatica*.
Infatti, se in una base di conoscenza Giovanni è codificato come persona e come
partecipante a un turno di notte in un B&B, Giovanni sarà automaticamente

classificato come CustodeNotturnoDiB&B. L'interesse di questo tipo di classificazione per applicazioni che intendono aggregare dati complessi e distribuiti è facilmente intuibile, se si esce dall'esempio giocattolo e si estende alle inferenze necessarie in ambiti realistici, come la cooperazione applicativa, i cruscotti aziendali, il riconoscimento di situazioni, etc.

3.1.4 I contenuti delle ontologie

Dal punto di vista dell'ingegneria ontologica, un contenuto è una qualsiasi struttura di conoscenza che si vuole rendere esplicita. Tale conoscenza può riguardare oggetti, processi, idee, modi di essere, di fare o di dire, leggi, metodi e praticamente qualsiasi cosa la mente umana possa concepire. D'altra parte, proprio per la varietà dei sistemi di riferimento e i modi potenzialmente infiniti di interagire con essi, è molto difficile immaginare *standard di contenuto*. Per esempio, nel caso del CustodeNotturnoDiB&B sono disponibili altre soluzioni di modellazione ontologica, soprattutto se viene adottato un linguaggio più espressivo. Ma quand'anche si elencassero tutti le possibili soluzioni efficienti, rimarrebbe la difficoltà di ingabbiare la specificità del caso in esame. Per esempio, il caso del CustodeNotturnoDiB&B sembra abbastanza neutrale, ma cosa succederebbe se quel caso fosse parte dei requisiti di un'applicazione per i test sulle procedure di sicurezza nei B&B? o per realizzare dei profili-cliente? o per parlare di un film in cui si raccontava di un CustodeNotturnoDiB&B? Probabilmente, il profilo-cliente o il testing delle procedure richiederebbe un livello di complessità maggiore (per modellare "tipi astratti" di eventi, cioè modi di fare qualcosa), mentre il film richiederebbe una complessità legata a problemi estetici più che fattuali.

Eppure, in molti domini, quando c'è un vero interesse (economico, politico, sociale), è stato già possibile creare dei *contenuti condivisi* (per esempio nell'area dei prodotti industriali, dei processi di lavoro, dei regolamenti sociali, vedi scheda 3). Il raggiungimento di tale condivisione è però laborioso, proprio per la varietà dei modi di interagire fra agenti e sistemi di riferimento e per la difficoltà di inventare *schemi di comunicazione* che permettano l'esplicitazione dei punti di vista degli agenti che creano quei contenuti condivisi. In passato, la conoscenza è stata condivisa in grandissima parte mediante repertori informali: manuali, documentazione, dizionari, enciclopedie, tesauri, glossari, schemi improvvisati, trasmissione orale, *know-how* personale distribuito. Una parte delle conoscenze è stata così ben isolata dai sistemi di riferimento e dai punti di vista degli agenti che una sua formalizzazione stabilmente riconosciuta si è resa possibile. Matematica e logica sono i risultati più evidenti di questa operazione. E infatti a questi modelli di formalizzazione si è rivolta l'ingegneria ontologica per definire i suoi linguaggi e cercare di codificare contenuti che però non sono facilmente circoscrivibili e isolabili dal loro contesto. Le difficoltà specifiche dell'ingegneria ontologica nascono in questo scenario.

In ingegneria ontologica, ci sono quattro modi principali di accostarsi alla

definizione di contenuti⁷:

1. Concettualizzazione locale e distribuita. In questo modo, un agente isolato, un gruppo, o una comunità *si mette d'accordo su un'ontologia rispondente ai propri requisiti*. Il vantaggio è la precisione con cui si formalizzano i requisiti locali. Lo svantaggio, è la difficoltà di raggiungere un accordo condiviso, nonché la difficoltà di mappare quest'ontologia con un'altra costruita in un dominio simile, ma con criteri diversi.
2. Reingegnerizzazione di conoscenza già condivisa. In questo modo, *si trasforma un repertorio informale* in un'ontologia seguendo linee-guida specifiche (il Semantic Web Best Practices and Deployment Working Group del W3C: <http://www.w3.org/2001/sw/BestPractices/> ha prodotto linee-guida per la reingegnerizzazione; il progetto NeOn: <http://www.neon-project.org>, sta estendendo questo approccio). Le strutture di dati che più tipicamente si reingegnerizzano sono le seguenti:
 - Linguaggi di modellazione: E/R, UML, XSD, Petri Nets, ebXML, BPEL4WS. In questo caso, la reingegnerizzazione consiste nel costruire una tavola di traduzione fra i costrutti usati in un linguaggio di modellazione, e i costrutti logici. Spesso questa operazione non è semplice e le mappature richiedono scelte, raffinamenti, approssimazioni.
 - Modelli concettuali: schemi di database, diagrammi UML, schemi XSD, etc. In questo caso, la reingegnerizzazione consiste nell'applicare una tavola di traduzione e le eventuali buone pratiche per ottenere il risultato ottimale.
 - Strutture di dati informali: Spreadsheets, tabelle, etc. In questo caso, la reingegnerizzazione consiste nell'indurre o nello stabilire ex-novo la semantica delle dimensioni con le quali sono state costruite le tabelle e quindi importare i dati contenuti secondo la semantica scelta.
 - Risorse lessicali: WordNet, FrameNet, Oxford Dictionary, etc. In questo caso, la reingegnerizzazione consiste nel costruire o applicare un meta-modello che integri dati lessicali ed elementi ontologici. In alcuni casi, è possibile tradurre direttamente il modello della risorsa lessicale in un modello ontologico. Per esempio, gran parte delle applicazioni di ontology learning (3.1.5) usano le risorse lessicali come se fossero ontologie, aggiungendo così un grado di approssimazione alle tecniche già approssimate che si usano.
 - Concept Schemes: Thesauri, classificazioni, nomenclature, etc. In questo caso, la reingegnerizzazione consiste nel costruire o applicare un meta-modello che integri elementi dello schema di concetti ed elementi ontologici. Un esempio di questi meta-modelli è SKOS.

⁷ Il progetto NeOn (Networked Ontologies, <http://www.neon-project.org>) sta studiando estesamente le tecniche per la definizione dei contenuti in ingegneria ontologica.

- Risorse Web 2.0: Folksonomies (ex. tag da Flickr o de.li.ci.o.us), categorie da Wikipedia, subject directories, etc. In questo caso, la reingegnerizzazione consiste nel formulare delle ipotesi sulla semantica dei tag e poi procedere a mapparla sul meta-modello di un linguaggio per ontologie.
- Documenti e corpora in linguaggio naturale. In questo caso, la reingegnerizzazione consiste nel condurre un vero proprio apprendimento (più o meno automatico). Vedi anche 3.1.5.

Il vantaggio della reingegnerizzazione è che si parte da qualcosa di usato e già condiviso. Lo svantaggio è la poca precisione con cui tali repertori mappano di solito le conoscenze, nonché la difficoltà di analizzare le fonti e di convivere con repertori che hanno già un loro ciclo di vita.

3. Progettazione modulare e/o stratificata. In questo modo, *si adattano componenti ontologici esistenti*, ma più generali, al dominio da formalizzare. Per esempio, in un dominio legale, si può riusare un'ontologia già sviluppata per la formalizzazione delle leggi, sempre che questa si adatti ai requisiti e agli obiettivi dell'ontologia. Altri esempi: un'ontologia che contenga la formalizzazione dei sistemi di misura, un'altra che descriva la struttura di eventi fisici, o i componenti di un tipo di sistema, o i flussi di lavoro in un'industria manifatturiera, etc. Alcune ontologie addirittura contengono la formalizzazione di nozioni talmente generali da essere praticamente indipendenti dal dominio di applicazione (ontologie *fondazionali*). Il vantaggio del riuso sta nel risparmio di tempo, nella possibilità di partire da schemi che qualcun altro ha già condiviso, nella possibilità di fornire *design pattern* per la costruzione di ontologie e, nel caso delle ontologie fondazionali, di *negoziare* la condivisione sulla base di criteri espliciti (laddove sono di solito molto difficili da esplicitare). Lo svantaggio è stato finora la mancanza di strumenti che rendano immediatamente usabili e comprensibili i componenti da riusare (un repertorio di componenti riusabili è in via di sviluppo al Laboratorio di Ontologia Applicata dell'ISTC-CNR, v. <http://www.loa-cnr.it/codeps>).
4. Apprendimento automatico. Il fine dell'apprendimento automatico di ontologie è quello di *estrarre le strutture* che possono far pensare a strutture di conoscenza condivise emergenti da grandi masse di dati (perché ricorrenti o rilevanti per altri motivi). Tipicamente, si cerca di estrarre *pattern* da testi in linguaggio naturale, ma anche oggetti da immagini, dati da basi di dati distribuite, etc. In particolare, l'apprendimento automatico da testo espresso in linguaggio naturale apre orizzonti incoraggianti per l'ingegneria ontologica, in quanto l'enorme disponibilità di materiale testuale presente sul web consentirebbe, se le tecnologie fossero ad uno stato maturo, la rappresentazione dei contenuti condivisi da comunità amplissime, sgravando notevolmente, o nella migliore delle ipotesi eliminando del tutto, l'intervento manuale di esperti di dominio, tipicamente molto oneroso. I vantaggi legati all'utilizzo di queste metodologie in ingegneria ontologica

sono dunque l'economicità e l'oggettività delle strutture apprese, mentre il principale svantaggio, a oggi, è la notevole imprecisione degli algoritmi, che nella maggior parte dei casi si limitano a fornire materiale “predigerito” agli esperti di un dominio, suggerendo nuove istanze identificate nei testi, nuove relazioni e talvolta inducendo tassonomie di concetti, in ogni caso con gravi lacune di copertura e pecche di affidabilità.

Nonostante il gran discutere sui pro e i contro di ognuno di questi metodi, la pratica dell'ingegneria ontologica è tutta a favore della complementarità dei quattro modi: in alcuni domini e per certi obiettivi converrà usare uno, alcuni o anche tutti i modi. Per esempio: i) i prodotti semi-lavorati della reingegnerizzazione e dell'apprendimento automatico vanno di solito raffinati mediante accordi locali o allineamenti a componenti riusabili; ii) la condivisione locale ha spesso bisogno di negoziazione mediante criteri generali o ha bisogno di partire dall'evidenza di un apprendimento automatico o di un repertorio esistente; iii) la progettazione modulare deve agire su ontologie pre-lavorate. Nell'esempio del CustodeNotturnoDiB&B, abbiamo seguito il metodo (1), ma abbiamo anche usato alcune nozioni e metodi preesistenti, facenti parte di *design pattern* collaudati (metodo (3)). Nelle prossime sezioni, daremo qualche dettaglio in più sui metodi (3) e (4).

Gli *ontology design patterns*

Al fine di realizzare un'ontologia in modo compiuto, occorre formalizzare i contenuti secondo uno o più linguaggi formali dedicati, sia per la definizione delle ontologie di dominio, sia per la definizione dei servizi che utilizzano l'ontologia. Nell'esempio del CustodeNotturnoDiB&B, questo tipo di attività corrisponde alla creazione dell'ontologia secondo uno o più modelli di progettazione adatti al task/contesto e alla definizione dei servizi relativi: nell'esempio si è usato il linguaggio OWL(DL), il modello di progettazione *Oggetto>partecipazione<Evento*, mentre l'unico servizio menzionato è la classificazione automatica di individui, la cui funzionalità è garantita da “ragionatori automatici” come FaCT (<http://www.cs.man.ac.uk/~horrocks/FaCT/>), Pellet (<http://www.mindswap.org/2003/pellet/>), PowerLoom (<http://www.isi.edu/isd/LOOM/PowerLoom>), etc.

I modelli di progettazione ontologica o *ontology design patterns* (Gangemi, 2005) permettono di risparmiare tempo ed errori utilizzando distinzioni già pensate e sperimentate in altri domini. Come esempio includiamo qui due modelli di progettazione, di cui il primo è stato parzialmente usato nella modellazione dell'esempio, mentre l'altro è usato nella modellazione di contesti e conoscenza sociale e sarà qui esemplificato con un esempio bancario. I modelli di progettazione sono visualizzati qui seguendo un codice iconico noto come *diagrammi di classi* nel linguaggio UML (*Unified Modelling Language*, <http://www.uml.org>). Un diagramma di classi rappresenta una classe con un rettangolo, una relazione binaria con una linea congiungente due rettangoli, la quantificazione applicabile con cardinalità sulle linee e l'implicazione logica (“superclasse” o sussunzione) con una freccia che termina in un triangolino. UML è molto usato nella

progettazione di sistemi informativi perché permette la verifica e generazione automatica di codice software. Il suo uso per visualizzare ontologie è dunque piuttosto naturale.

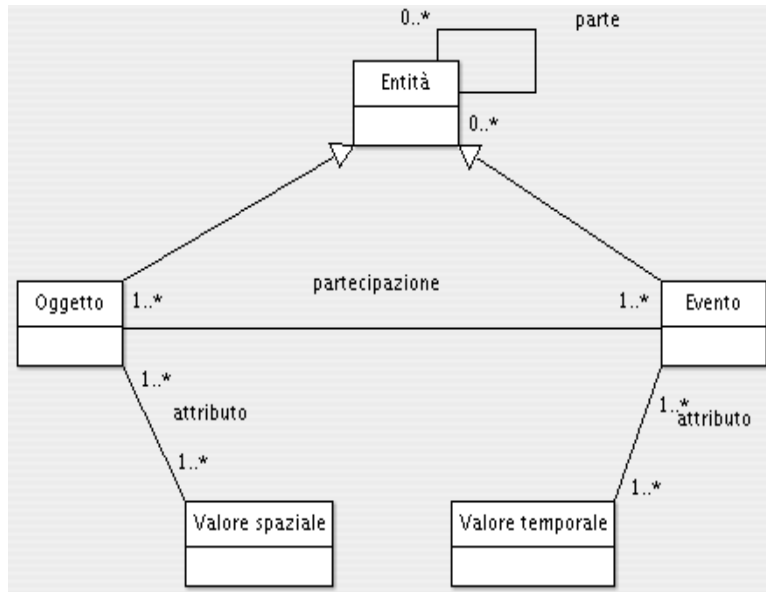


Figura 3.2: Il pattern *Oggetto>partecipazione<Evento*

Il semplice *design pattern* *Oggetto>Partecipazione<Evento* (<http://www.loa-cnr.it/codeps/pcp>), visualizzato in Fig. 3.2, è parte dell'ontologia fondazionale DOLCE (<http://dolce.semanticweb.org>) e caratterizza oggetti ed eventi come tipi di entità e gli oggetti come i partecipanti degli eventi, secondo un'assiomatizzazione particolare, che coinvolge anche la relazione di parte, i valori spaziali e temporali, etc. Questo pattern permette di parlare di eventi qualsiasi con un numero di tipi di partecipanti illimitato (vedi esempio del mangiatore notturno). L'applicazione del pattern, che eredita l'assiomatizzazione collaudata di DOLCE, esclude errori logici e suggerisce la ricerca di nuovi – o la modellazione di altri – componenti di dominio (ex. date come valori temporali, coordinate come valori spaziali, etc.). *Oggetto>Partecipazione<Evento* è facilmente componibile con altri modelli di progettazione, ma anche con se stesso, producendo il modello *Co-Partecipazione*, che è molto utile per rappresentare gran parte delle attività fisiche e sociali, in quanto queste richiedono di solito di rappresentare oggetti di tipo diverso che partecipano tutti a uno stesso evento (ex. persone che preparano cibo). Se *Co-Partecipazione* viene poi composto con *Oggetto>classificazione<Ruolo*, è possibile assegnare ruoli diversi agli oggetti che co-partecipano a un evento, raggiungendo quindi l'espressività richiesta per esempio dalle grammatiche tradizionali del linguaggio naturale.

Un esempio più complesso di *ontology design pattern*, che va oltre l'espres-

sività delle grammatiche tradizionali e permette di analizzare processi sociali usualmente rappresentati con costrutti linguistici più complessi è *Descrizione* ↔ *Situazione* (Fig. 3.3), parte dell'ontologia D&S ("Descriptions and Situations", <http://www.loa-cnr.it/codeps/dns>) ed è usato per modellare contesti, piani, linee-guida, obiettivi, regole, norme, etc. E' particolarmente appropriato a rappresentare conoscenza interpretativa o di controllo su dati distribuiti e da aggregare secondo modelli predefiniti o emergenti (il problema di rappresentare l'interazione sociale, per esempio nel dominio della pianificazione e della gestione delle organizzazioni, può essere ricondotto a questa caratterizzazione). Questo modello di progettazione contiene una soluzione pre-modellata e facilmente specializzabile per parlare di *situazioni* contestuali secondo un certo punto di vista (chiamato *descrizione*). Una descrizione tipicamente *definisce ruoli, strutture di eventi e parametri* che permettono di *classificare* oggetti, eventi e valori rispettivamente: quando la classificazione riesce a estrarre dati coerenti da un insieme, emerge una situazione, che risulta quindi essere una buona *contestualizzazione* per le entità classificate. In questo caso, diciamo che la situazione *soddisfa* la descrizione.

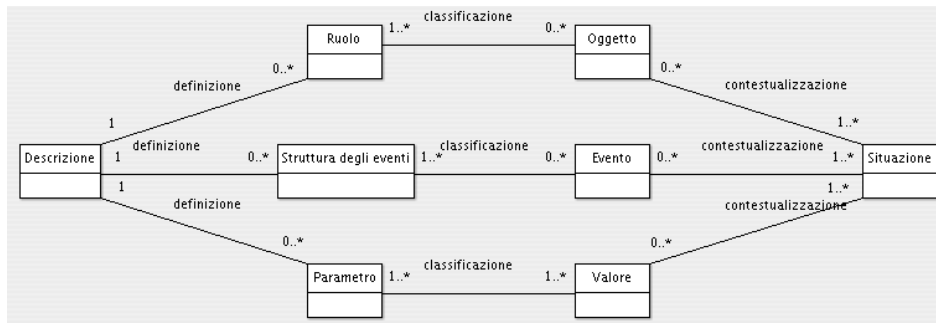


Figura 3.3: Il pattern *Descrizione* ↔ *Situazione*

Per esempio, nel mondo delle transazioni bancarie, si può rappresentare come una transazione bancaria soddisfi (o non soddisfi) una linea-guida per la scoperta di transazioni anomale, per esempio una situazione di riciclaggio. Gli elementi di una *situazione* bancaria rendono tale situazione anomala se globalmente soddisfano gli elementi della *descrizione* di anomalia: per esempio una situazione costituita da una persona che svolge una certa attività e svolge un'operazione bancaria spostando una certa quantità di denaro sarà anomala se la persona svolge il *ruolo* di cliente di una certa banca, se la sua attività e tipo di operazione rientrano in una *struttura degli eventi* stabilita dalla linea-guida e se la quantità di denaro spostata rientra nei *parametri* stabiliti dalla linea-guida. Come è intuibile, la modellazione di linee-guida e situazioni in questo caso non è semplicissima, non ha una codifica lessicale tipica e richiederebbe una lunga analisi semantica e un'ardua formalizzazione. Il pattern consente di abbreviare il percorso di analisi e fornisce un'assiomatizzazione già sperimentata.

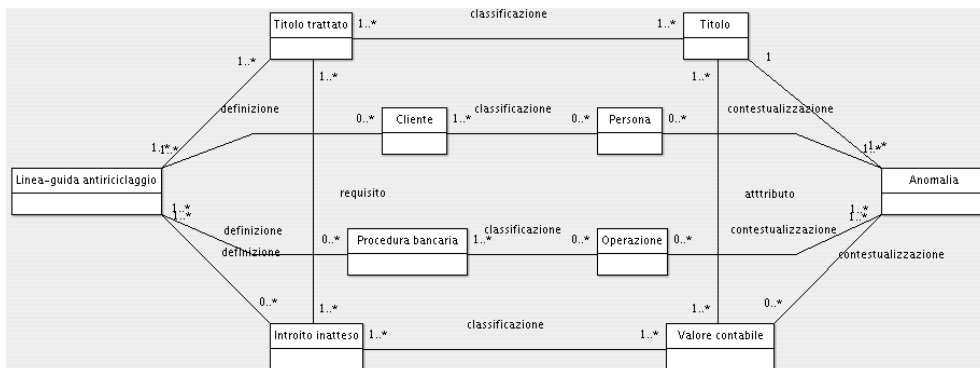


Figura 3.4: Il pattern *Descrizione*↔*Situazione* specializzato per la modellazione di scenari di anomalia nelle operazioni bancarie

3.1.5 Apprendimento di ontologie da testi

Uno dei limiti principali del web semantico, almeno nella sua visione globale, è costituito dall'impossibilità di realizzare per via del tutto automatica le meta-informazioni necessarie affinché i contenuti nei testi siano esplicitati e condivisi sia da comunità di utenti sia da agenti software. L'utente è tipicamente pigro, l'informazione viene solitamente prodotta e consumata in fretta, rendendo pressoché vana la richiesta per un ulteriore sforzo di concettualizzazione. Gran parte dell'informazione oggi presente su web continua ad essere espressa in linguaggio naturale.

Pertanto, è stato oggetto di crescente interesse il problema dell'acquisizione automatica di ontologie da testo. L'interesse in questa area è motivato dal fatto che le ontologie costituiscono lo strumento principe per la rappresentazione della conoscenza per il web semantico, conoscenza il più delle volte contenuta solo come informazione nei testi.

Il quadro di riferimento teorico di queste ricerche è la linguistica computazionale, ovvero l'analisi delle proprietà delle lingue naturali mediante la definizione di teorie algoritmiche implementabili da calcolatori, oggetto del presente volume. L'applicazione tecnologica di tali studi è oggetto d'interesse del settore dell'intelligenza artificiale noto come *elaborazione del linguaggio naturale* (NLP), che si occupa di sviluppare sistemi intelligenti per l'analisi del testo. Come illustrato nei capitoli relativi alla sintassi ed alla semantica contenuti nel presente volume, i sistemi per l'elaborazione del testo riescono ad assolvere numerosi compiti tipicamente connessi ai problemi della comprensione del testo, come ad esempio la categorizzazione automatica, l'analisi sintattica, l'estrazione di informazione e molto altro. Le tecnologie sviluppate in questo ambito compongono la base per lo sviluppo dei sistemi per il recupero automatico dell'informazione. L'impatto sociale di queste tecnologie è quindi enorme, in quanto esse agevolano il quotidiano svolgimento delle operazioni di miliardi di utenti.

L'insieme di testi messo a disposizione dall'avvento del web ha suggerito a

più voci la necessità, o meglio il desiderio, di acquisire la conoscenza richiesta da corpora documentali di vaste dimensioni. Più recentemente, le ontologie sono divenute le forme di rappresentazione privilegiate per la rappresentazione della conoscenza sul web. Pertanto, la disciplina nota come *Ontology Learning from Texts* (Buitelaar et. al, 2005), ha attratto un notevole interesse da parte della comunità scientifica, manifestato dall'organizzazione di numerosi workshop ed eventi internazionali. Ad oggi, tale disciplina ha prodotto un insieme di tecnologie alquanto variegata. Infatti, come illustrato nella sezione precedente, la nozione stessa di ontologia si presta a numerose interpretazioni, caratterizzate da gradi sempre crescenti di precisione. Se, infatti, si intende per ontologia un semplice tesauro o tassonomia, le tecnologie di elaborazione automatica del testo richieste per l'acquisizione automatica saranno certamente più semplici se paragonate a quelle richieste per l'acquisizione di una ben più complessa teoria assiomatica, che prevede la descrizione di relazioni e vincoli fra entità.

In tutti i casi, delineare il dominio di interesse, e quindi la collezione di testi da cui estrarre i contenuti, è un passo fondamentale per la realizzazione di un processo automatico di costruzione di ontologie. Tale operazione può essere effettuata utilizzando sistemi *supervised* per la categorizzazione automatica del testo, tecniche basate su *relevance feedback* per il recupero dell'informazione o approcci basati su *bootstrapping* da parole-chiave per la categorizzazione del testo (Gliozzo et. al, 2005b). In uno scenario multilingua, inoltre, può essere utile collezionare corpora in lingue differenti facenti riferimento ad un medesimo dominio, ad esempio mediante l'utilizzo di sistemi multilingua per la categorizzazione del testo (Gliozzo e Strapparava, 2006). Di tali tecnologie si è ampiamente discusso nel capitolo relativo alla semantica, contenuto nel presente volume.

Una volta stabilito il dominio di interesse, l'apprendimento automatico procede attraverso l'utilizzo di metodologie per l'acquisizione di conoscenza a vari livelli, come illustrato graficamente in Fig. 4. A livelli crescenti corrispondono algoritmi di complessità crescente, il cui stato dell'arte è lontano dall'aver trovato una soluzione definitiva.

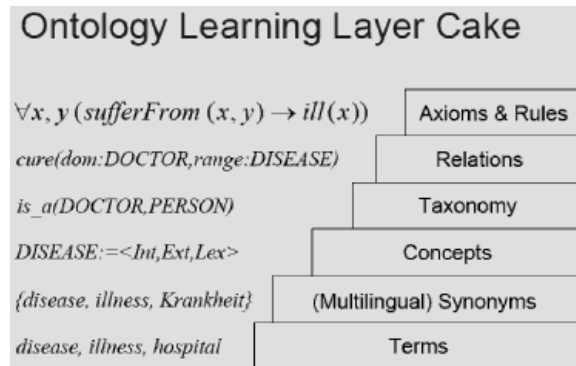


Figura 3.5: Livelli di elaborazione del testo richiesti per l’acquisizione automatica di ontologie

Al primo livello corrispondono i sistemi per l’*estrazione automatica di terminologia*, che forniscono il materiale essenziale necessario alla costruzione dei concetti di una ontologia: le loro espressioni linguistiche. In genere tali sistemi sono realizzati integrando componenti per l’analisi sintattica finalizzati al riconoscimento dei sintagmi nominali e componenti statistici il cui scopo è stabilire il grado di associazione tra le parole che compongono il sintagma. Per esempio, i sintagmi caratterizzati dal pattern sintattico nome + aggettivo (*chimica industriale*, *linguistica computazionale*) e nome + preposizione + nome (*Elaborazione del testo*, *base di conoscenza*) sono ottimi candidati. Al fine di selezionare solo termini rilevanti per il dominio in analisi, in genere si utilizzano misure statistiche per stabilire il grado di associazione tra i termini che compongono il sintagma e misure distribuzionali per determinare la rappresentatività del termine nel suo complesso, ad esempio selezionando solo i termini caratterizzati da una bassa entropia.

Al livello successivo si collocano le metodologie per il *riconoscimento dei sinonimi*, al fine di identificare un insieme di varianti lessicali con il quale il medesimo concetto può essere espresso, e quindi identificato. In genere la tecnica utilizzata è la misura della similitudine distribuzionale (Dagan, 2000), consistente nell’analisi delle similitudini tra i contesti in cui le parole sono collocate all’interno di corpora di grandi dimensioni. Se formulato in uno scenario multilingua, il medesimo problema può essere descritto come il compito di identificare possibili traduenti. A tal proposito, le tecniche che assicurano le migliori prestazioni sono basate sull’utilizzo di corpora paralleli. Tuttavia, tali risorse sono difficoltosamente reperibili per molti domini. Pertanto la ricerca attuale tende a concentrarsi sull’utilizzo di corpora comparabili (Gliozzo e Strapparava, 2006).

La popolazione della base di conoscenza consiste nell’inserimento di costanti individuali in funzione dell’ontologia. Ad esempio, al concetto di *sciatore* saranno associate le istanze *Alberto Tomba* e *Kristian Ghedina*. Data un’ontologia di concetti preesistente, in genere, il popolamento automatico avviene mediante

l'utilizzo di tecnologie per il riconoscimento automatico delle entità nei testi. Allo stato dell'arte, le migliori prestazioni si possono ottenere mediante l'utilizzo di tecnologie supervisionate per l'annotazione semantica (Gliozzo et. al., 2005a). Il limite principale di tali approcci è che essi richiedono una notevole quantità di materiale di addestramento annotato manualmente. Pertanto tali tecnologie sono in genere utilizzate per riconoscere entità appartenenti a concetti generali, come ad esempio *persone*, *luoghi* ed *organizzazioni*, mentre essi si rivelano inadeguati per il riconoscimento di istanze di concetti di più basso livello, come *scienziati* o *sciatori*, nell'esempio precedente. Per quest'ultimo compito, è possibile utilizzare tecnologie basate su *bootstrapping* da esempi (Fleischman e Hovy, 2002) o metodologie basate, ancora una volta, sul riconoscimento della similitudine distribuzionale (Cimiano e Völker, 2005).

Se si può di buon grado sostenere che ad oggi esistono tecnologie efficienti per la soluzione dei problemi suddetti, altrettanto non è asseribile in relazione ai livelli superiori del processo di acquisizione. In genere, le tecnologie che ottengono i migliori risultati (seppur modesti) nell'affrontare problemi come l'*induzione di tassonomie* sono basate su approcci integrati, che da un lato utilizzano risorse linguistiche preesistenti, come ad esempio WordNet, e dall'altro inferiscono dal corpus la rilevanza di alcuni concetti a discapito di altri (Basili et. al., 2006), determinando sottostrutture coerenti con il dominio di interesse.

Una volta acquista una tassonomia che include i concetti dell'ontologia e popolata mediante istanze di concetti, è spesso necessario identificare le relazioni che intercorrono tra le istanze, che corrisponde logicamente ad identificare asserzioni concernenti coppie di individui. Anche in questo caso, i migliori risultati possono essere ottenuti mediante l'utilizzo di tecnologie supervisionate per l'estrazione di relazioni da testi. Tuttavia, la quantità di materiale annotato necessario per addestrare tali sistemi è spesso difficilmente producibile per le relazioni di interesse. Pertanto, approcci basati sul riconoscimento e il bootstrapping di *pattern* lessico-sintattici sono preferibili in uno scenario di acquisizione totalmente automatico (Gliozzo et al., 2007).

Infine, l'ultimo livello della gerarchia sopra descritta prevede lo sviluppo di sistemi finalizzati all'inferenza di regole e assiomi da testo. Questo campo è stato scarsamente esplorato, ma sta divenendo oggetto di sempre crescente interesse in letteratura, in quanto l'acquisizione di regole di inferenza è certamente un aspetto fondamentale legato al riconoscimento dell'implicazione nel testo (Dagan et al., 2005). Inoltre l'identificazione di tali proprietà è di importanza cruciale per attivare processi di ragionamento automatico. Un esempio di acquisizione di relazioni e pattern ontologici da pattern testuali è (Ciaramita et al., 2005).

Ad oggi, i sistemi disponibili per la costruzione automatica di ontologie da testi sono in realtà piattaforme di sviluppo che agevolano la costruzione manuale di tali artefatti, integrando al loro interno componenti per l'acquisizione di conoscenza e fornendo un materiale di base da far valutare manualmente all'utente. La maggior parte di questi componenti sono disponibili in modalità *open source* e sono in grado di generare una rappresentazione OWL della conoscenza acquisita. Elenchiamo di seguito alcuni strumenti che possono essere adottati a

questo scopo ed i puntatori agli indirizzi web presso i quali è possibili scaricarli gratuitamente:

- Ontolearn - <http://lcl.di.uniroma1.it/tools.jsp>
- Text2Onto - <http://ontoware.org/projects/text2onto/>
- OntoGen - <http://ontogen.ijs.si/>
- Gate - <http://gate.ac.uk/>

Tuttavia, molta strada deve essere ancora effettuata affinché un processo pienamente automatizzato che soddisfi i requisiti richiesti dall'ingegneria ontologica sia effettivamente realizzato.

3.1.6 Qualità di un'ontologia

La questione della qualità di un'ontologia è attualmente molto dibattuta e non si ha un benchmark né una lista di criteri standard; anzi, sembra che i criteri e i vincoli sulle misure possano variare in scenari diversi (Gangemi et al., 2006). I criteri attualmente più promettenti sono i seguenti.

La *copertura* è la misura della quantità di modelli desiderati rappresentabili da un'ontologia; è ovviamente complementare alla *specificità*. Per esempio, l'ontologia dei custodi notturni di b&b è molto specifica, perché permette di parlare di persone che fanno il turno di notte in un b&b, ma non di pulitori d'albergo che lavorano in b&b di giorno. Una copertura ampia permette di parlare di molte cose, ma difficilmente l'ontologia sarà precisa, oppure dovrà essere molto complessa e quindi di difficile sostenibilità (vedi sotto). Una copertura specifica ha caratteristiche inverse: l'ontologia può essere molto precisa ma richiede estensioni ogni volta che occorre esprimere qualcosa di non specifico. In genere, occorrono sia ontologie ad ampia copertura, in una prospettiva di *riusabilità* (cf. la sezione 5. Sugli *ontology design pattern*), sia ontologie specifiche.

La copertura è una misura “funzionale” e come tale, di difficile calcolo, perché la sua misura perfetta richiede proprio quello che si vuole costruire, cioè lo spazio semantico di riferimento dell'ontologia. Per tale motivo, le misure di copertura sono tutte indotte a partire da altri tipi di dato, di solito corpora testuali o domande “di competenza” rivolte agli esperti del dominio del quale l'ontologia dovrebbe coprire i modelli attesi.

La *precisione* è la misura della selettività di un'ontologia: quanti dei modelli rappresentabili in un'ontologia sono effettivamente voluti? Per esempio, l'ontologia dei custodi notturni di b&b permette di parlare di persone che fanno il turno di notte in b&b, ma, poiché non contiene una restrizione che vincoli il tipo di azienda, anche un portiere di notte in alberghi a cinque stelle rientrerebbe nei modelli possibili. Anche la precisione richiede un approccio di misurazione simile alla copertura.

Il *dettaglio* assiomatico è la misura della quantità di assiomi che sono presenti in media per ogni classe nell'ontologia ed è complementare alla precisione

e all'*espressività*, che consiste nella complessità delle strutture logiche impiegate nella costruzione degli assiomi. Il dettaglio assiomatico è misurabile in base all'ontologia stessa (è una misura "strutturale"), ed è una misura molto buona della qualità di un'ontologia, laddove non si abbiano problemi di *sostenibilità*. La sostenibilità misura quanto un'ontologia può essere di fatto usata e mantenuta da un'organizzazione; per esempio, un'ontologia con un alto dettaglio assiomatico, alta precisione, ampia copertura, è in genere piuttosto onerosa da progettare, costruire, mantenere e soprattutto usare in sistemi informativi con molti dati, perché la quantità di inferenze generate dai ragionatori automatici attuali aumenta in funzione delle misure suddette.

La *flessibilità* misura per esempio se sia possibile rappresentare punti di vista diversi sulle stesse entità del dominio e se sia possibile rappresentare annotazioni sulla "fiducia" del modellatore in una certa distinzione o assioma. Questa misura è permessa solo da ontologie che contengano informazione sufficiente sul loro *design*, cioè su come sono state progettate, con quali componenti, con quali criteri e processi argomentativi e decisionali, da chi, per cosa sono state usate, etc.

Copertura, precisione, dettaglio e flessibilità non sono necessariamente correlate: alcune ontologie altamente riusabili ("ontologie di riferimento") hanno un'espressività molto elevata, altre no; alcune ontologie "di dominio" sono molto dettagliate, altre no. La mancanza di correlazione è dovuta probabilmente agli diversi scenari di applicazione e sostenibilità: le ontologie di dominio possono essere poco assiomatizzate per applicazioni stile semantic web ed estrazione d'informazione (almeno per ora), ma possono essere molto assiomatizzate per applicazioni di supporto alla decisione. Inoltre, le ontologie di dominio possono essere a volte altamente riusabili (fanno da riferimento per un dominio, per esempio nella medicina), dedicate a una classe di applicazioni (per esempio nel controllo distribuito di dati epidemiologici) o dedicate a uno specifico database (per esempio nella gestione degli interventi clinici di un certo comprensorio).

3.1.7 Strumenti software

Attualmente la disponibilità di strumenti software per l'ingegneria ontologica è piuttosto limitata. Quasi tutti sono stati progettati pensando a utenti esperti e con requisiti prevalentemente logici. Requisiti come *assistenza alla progettazione del contenuto*, *scalabilità* e *usabilità dell'interfaccia utente* sono stati finora pressoché ignorati.

Recentemente, alcuni progetti e aziende hanno avviato attività di sviluppo rispetto a questi requisiti, come testimoniano per esempio i progetti NeOn: <http://www.neon-project.org>, co-ode: <http://www.co-ode.org> e CBio: <http://bioontology.org>, l'azienda TopQuadrant: <http://www.topquadrant.com>, il workshop CKC: <http://km.aifb.uni-karlsruhe.de/ws/ckc2007> e il portale <http://www.ontologydesignpatterns.org>.

3.1.8 Letture consigliate

Per reperire materiale di riferimento a largo raggio e cominciare a orientarsi nell'ormai imponente letteratura e documentazione sulle ontologie, si raccomandano, oltre agli indirizzi web già citati nel testo, i portali del Laboratorio di Ontologia Applicata del CNR-ISTC: <http://www.loa-cnr.it> e <http://dolce.semanticweb.org>, della rete di eccellenza KnowledgeWeb: <http://knowledgeweb.semanticweb.org>, dei progetti WonderWeb: <http://wonderweb.semanticweb.org>, SEKT: <http://sekt.semanticweb.org>, DIP: <http://dip.semanticweb.org>, NeOn: <http://www.neon-project.org>, OntoText: <http://tcc.itc.it/projects/ontotext/>, DAML: <http://www.daml.org>, del dipartimento di *computer science* dell'Università di Stanford: <http://ksl.stanford.edu/>, i portali dell'AIFB: <http://fzi.aifb.de>, del KMI: <http://kmi.open.ac.uk>, del laboratorio di *network dynamics* dell'Università del Maryland: <http://www.mindswap.org/>, della UPM: <http://delicias.upm.es> e del W3C: <http://www.w3.org>.

Publicazioni citate e rilevanti

Franz Baader (ed.). 2003. *The Description Logic Handbook: theory, implementation, and applications*. Cambridge University Press, Cambridge.

Roberto Basili, Marco Cammisa and Alfio Gliozzo. 2006. *Integrating Domain and Paradigmatic Similarity for Unsupervised Sense Tagging*. in Proceedings of ECAI-2006, Riva del Garda, Italy

Tim Berners-Lee, James Hendler and Ora Lassila. (2001). *The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities*. Scientific American, May.

Paul Buitelaar, Philippe Cimiano and Bernardo Magnini (eds.) 2005. *Ontology Learning from Text: Methods, Evaluation and Applications*, IOS press.

Massimiliano Ciaramita, Aldo Gangemi, Esther Ratsch, Isabel Rojas, Jasmin Saric. 2005. *Unsupervised Learning of Semantic Relations between Concepts of a Molecular Biology Ontology*, in Proceedings of International Joint Conference on Artificial Intelligence (IJCAI2005), Edinburgh, UK.

Philipp Cimiano and Johanna Völker. 2005. *Towards large-scale, open-domain and ontology-based named entity classification*. In Proceedings of RANLP'05, pages 66–166–172, Borovets, Bulgaria.

I. Dagan. 2000. *Contextual word similarity*. In Rob Dale, Hermann Moisl, and Harold Somers (eds.) , *Handbook of Natural Language Processing*, chapter 19, pages 459–476. Marcel Dekker Inc.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. *The pascal recognising textual entailment challenge*. In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. 1990. *Indexing by latent semantic analysis*. *Journal of the American Society of Information Science*.

Dekang Lin. 1998. *Automatic retrieval and clustering of similar words*. In Proceedings of ACL-98, pages 768–774, Morristown, NJ, USA.

- Michael Fleischman and Eduard Hovy. 2002. *Fine grained classification of named entities*. In Proceedings of ACL-2002, pages 1–7, Morristown, NJ, USA.
- Tom Gruber, 1993. *A translation approach to portable ontology specifications*. *Knowledge Acquisition*, 5, 199-220.
- Aldo Gangemi. 2005. *Ontology Design Patterns for Semantic Web Content*, In Proceedings of ISWC 2005, Galway, Ireland.
- Aldo Gangemi, Carola Catenacci, Massimiliano Ciaramita, Jos Lehmann. 2006. *Modelling Ontology Evaluation and Validation*, In Proceedings of ESWC 2006, Montenegro.
- Alfio Gliozzo. 2005. *Semantic Domains in Computational Linguistics*, Ph.D. thesis, University of Trento, December 2005.
- Alfio Gliozzo, Marco Pennacchiotti and Patrick Pantel. 2007. *Improving Relation Extraction Using Domain Information*. To appear in Proceedings of North American Association for Computational Linguistics / Human Language Technology (NAACL HLT 07). Rochester, NY
- Alfio Gliozzo and Carlo Strapparava. 2006. *Exploiting Comparable Corpora and Bilingual Dictionaries for Cross-Language Text Categorization*. In Proceedings of ACL 2006, Sydney, Australia.
- Alfio Gliozzo, Carlo Strapparava and Ido Dagan. 2005. *Investigating Unsupervised Learning for Text Categorization Bootstrapping*, in Proceedings of EMNLP-2005, Vancouver, B.C., Canada
- Chu-Ren Huang, Nicoletta Calzolari, Aldo Gangemi, Alessandro Lenci, Alessandro Oltramari, Laurent Prévot (eds.). 2007. *Ontologies and the Lexicon*. Cambridge University Press, Cambridge.
- Bernardo Magnini, Gabriela Cavaglià. 2000. *Integrating Subject Field Codes into WordNet*. In Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation, Athens, Greece
- D. McCarthy, R. Koeling, J.Weeds, and J. Carroll. 2004. *Finding predominant senses in untagged text*. In Proceedings of ACL-04, pages 280–287, Barcelona, Spain.
- Nicolas Nicolov, Franco Salvetti, Mark Liberman, and James H. Martin (eds.). 2006. Proceedings of the AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs. Stanford, CA.
- B. Pang and L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of ACL-04, Barcelona, ES.
- P. Pantel and M. Pennacchiotti. 2006. *Espresso: A bootstrapping algorithm for automatically harvesting semantic relations*. In Proceedings of COLING/ACL-06.
- R. Snow, D. Jurafsky, and A.Y. Ng. 2006. *Semantic taxonomy induction from heterogenous evidence*. In Proceedings of the ACL/COLING-06, pages 801–808, Sydney, Australia.
- Steffen Staab and Rudi Studer (eds.). 2004. *Handbook on Ontologies*, Springer.

3.2 Esempi di ontologie riusabili e come impiegarle per l'integrazione semantica dei domini della PA

NOTA: questa sezione è ancora da completare.

Basato su materiale contribuito da Aldo Gangemi
CNR-ISTC e CNIPA

3.2.1 Introduzione

Lo scopo di questa sezione è fornire un esempio intuitivo di come si può controllare la qualità di un'ontologia sulla base di componenti riusabili e task e la sua integrazione con altre ontologie mediante analogie attraverso modelli condivisi.

La tecnica qui esemplificata usa i seguenti componenti:

1. Domande di competenza (*competency questions*)
2. Metamodelli per le risorse disponibili
3. Modelli di progettazione (*ontology design patterns*)
4. Regole di mappatura (*bridging rules*)

Con un esempio minimale spiegheremo di cosa si parla e come usare i componenti. La tecnica sarà poi ripresa nella sezione 5.1 con un esempio realistico e più ampio.

3.2.2 Domande di competenza, task e contesto dell'ontologia

La prima attività (Figura 3.6) per lo sviluppo di un'ontologia consiste nella definizione del task e del contesto. Il task è la descrizione dell'uso che si intende fare dell'ontologia, in particolare nella descrizione dei servizi (query, viste) che l'ontologia supporta (per mezzo di linguaggi e strumenti adeguati) per assistere un tipo di attività di uno o più agenti (questa stessa descrizione della metodologia di sviluppo di un'ontologia una rappresentazione semi-formale di un task complesso). Il contesto è lo scenario in cui i servizi dell'ontologia andranno svolti: tale scenario include l'ambiente software, i suoi utilizzatori e l'ambiente sociale nel quale il software e gli agenti che lo utilizzano si muovono. La definizione di task e contesto può essere fatta in maniera informale, oppure utilizzando linguaggi di specifica dei requisiti.

Aldilà del linguaggio di rappresentazione, è però fondamentale porre dei limiti alla specifica del task e del contesto. Il componente primario che permette di porre tali limiti in modo pratico ed efficiente consiste nell'esplicitare le *domande di competenza*.

Una domanda di competenza — una domanda che un esperto intende fare al sistema informativo nel quale l’ontologia dovrebbe avere un ruolo. Queste domande, rappresentabili anche come semplici query SQL su uno schema esistente o appena sbizzato, sono fortemente dipendenti dal contesto del sistema informativo e degli utenti e sono quindi un modo per rappresentare il *task* dell’ontologia da sviluppare. Per esempio, il responsabile del dipartimento BCNL del Ministero del Lavoro potrebbe essere interessato a *fornire agli utenti informazioni sulla distribuzione geografica della richiesta di competenze per rapporti di lavoro negli ultimi due anni*.

Un analista ontologico può estrarre questa domanda in vari modi (complementari):

- chiedendo al responsabile
- intervistando persone informate del dipartimento (per esempio, le cosiddette “*hub persons*” possono essere trovate con tecniche di *social network analysis*)
- analizzando gli schemi delle basi dati esistenti e confrontandole con un log di query effettuate
- estraendo conoscenza dai documenti relativi al funzionamento del dipartimento
- estraendo pattern ricorrenti dalla *log analysis* del sito web del dipartimento
- analizzando una lista di desiderata, etc.

La domanda può poi essere decomposta e rappresentata in vari modi (complementari anche questi):

- analizzando la struttura logica della domanda (anche mediante strumenti automatici)
- riportandola a una forma normale rispetto a un query language (ex. SPARQL), usando per esempio uno schema esistente che dovrà essere reingegnerizzato, o inventando un’ontologia preliminare
- specializzando un *caso d’uso generico* da un repertorio esistente (cf. <http://www.ontologydesignpatterns.org>)

Ad esempio, il caso sopra riportato potrebbe essere espanso come segue (il vocabolario dell’ontologia assunta è qui generato intuitivamente, per poi essere rivisto con esperti, schemi esistenti, documentazione, etc. Assumiamo data attuale=2007):

$$\begin{aligned}
 (x, y) \leftarrow & (Competenza(x) \wedge Luogo(y) \wedge Azienda(z) \wedge \\
 & RapportoLavoro(w) \wedge RichiestaCompetenze(k) \wedge Data(t) \wedge \\
 & presentaRichiesta(z, k) \wedge inData(k, t) \wedge haSede(z, y) \wedge \\
 & perCompetenza(k, x) \wedge perRapporto(k, w) \wedge \geq (t, 2005)) \quad (3.1)
 \end{aligned}$$

Una domanda di competenza del genere, attraverso gli opportuni passi di design (cf. Figura 3.6), porterebbe a stabilire un primo pezzetto di ontologia: quello necessario a costruire la query logica 3.1. Successive domande di competenza arricchirebbero poi l'insieme di elementi necessari a progettare la prima versione dell'ontologia.

Nella prossime sezioni, vediamo come questo esempio verrebbe trattato in funzione delle risorse e dei modelli di progettazione.

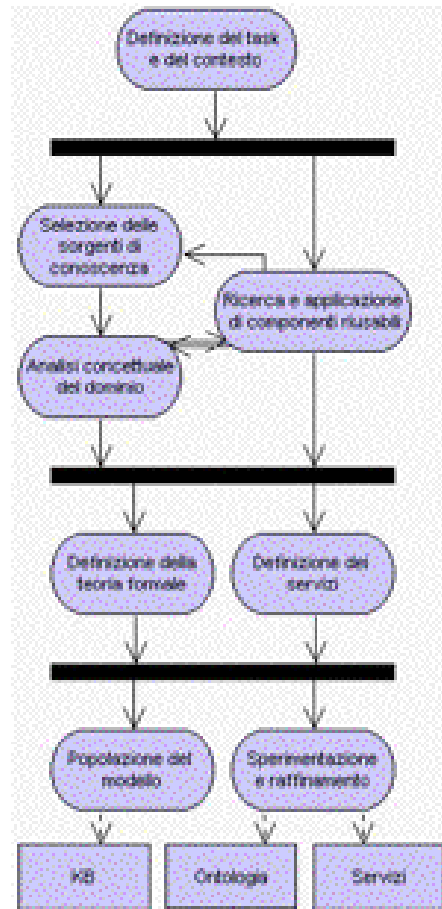


Figura 3.6: Una metodologia minimale per la progettazione di un'ontologia e del suo ruolo all'interno dei servizi

3.2.3 Selezione delle risorse disponibili e metamodelli di reingegnerizzazione

Come presentato in 3.1.4, molti tipi di sorgenti di conoscenza sono disponibili per la costruzione di ontologie. La loro selezione va effettuata di volta in volta in base ai requisiti dell'applicazione (per esempio, per la necessit di riusare una sorgente informale o essere interoperabili con essa), ai costi-benefici del riuso, etc.

A titolo di esempio, che sarà poi ripreso in 5, descriviamo un tipico workflow per la reingegnerizzazione di schemi XSD in ontologie, la loro gestione all'interno di un catalogo e il loro uso per effettuare query integrate (Fig. 3.7).

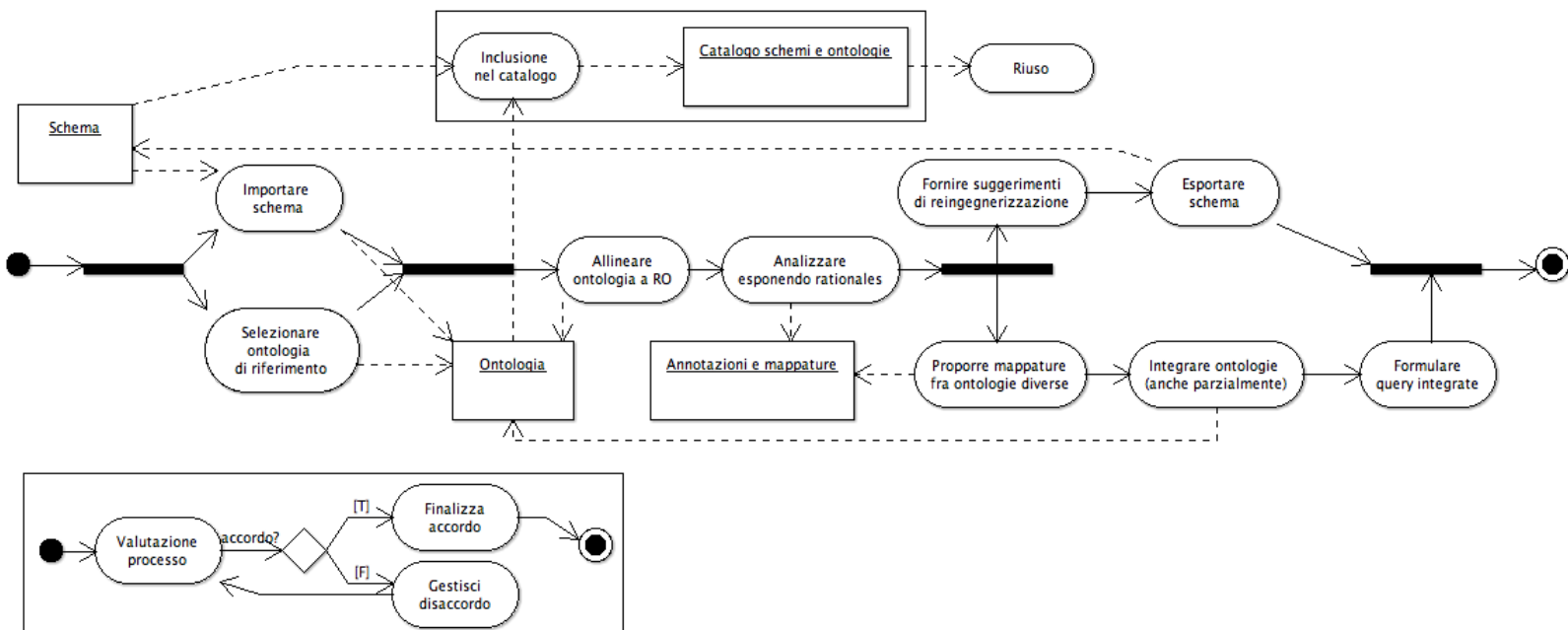


Figura 3.7: Un esempio di ciclo di vita per la reingegnerizzazione di schemi in ontologie, il loro uso e il loro inserimento in un catalogo

3.2.4 Ontologie fondazionali e di riferimento

3.2.5 Design guidato da modelli riusabili

3.2.6 Regole di mappatura e interoperabilità fra ontologie

3.3 Un'ontologia fondazionale per l'interoperabilità semantica nell'e-Government

Basato su materiale contribuito da:
Roberta Ferrario (CNR-ISTC, Trento)

3.3.1 Introduzione

Una delle maggiori sfide nell'e-Government quella di instaurare l'“interoperabilità semantica”, ovvero di far sì che l'informazione possa essere fruita e trasmessa mantenendo intatto il suo significato e riducendo al minimo errori e fraintendimenti. Data l'eterogeneità degli attori e delle strutture organizzative coinvolti nei vari processi e servizi, la sfida di difficile soluzione e le competenze che si richiedono per affrontarla esulano da quelle relative alla sola tecnologia.

I tentativi finora compiuti di dipanare la questione si sono perlopi rivolti alla formazione di consorzi di Pubbliche Amministrazioni (PA) in cui chi ha già implementato dei servizi di e-Government mette in condivisione la propria esperienza, per far emergere dal confronto delle best practices che possano essere assunte come standard e riutilizzate da tutte le PA partecipanti al consorzio. L'esperienza ha finora mostrato che le applicazioni sviluppate con tecnologie basate sulla semantica, come le ontologie, si rivelano particolarmente funzionali. La strategia del consenso di cui sopra, anche se sicuramente necessaria e auspicabile, non è da sola sufficiente a risolvere tutti i problemi. Essa infatti ancora in parte troppo centrata sulle applicazioni tecnologiche: nei casi in cui l'informazione debba essere trasmessa tra servizi diversi, non è sufficiente avere delle ontologie standard specifiche che descrivano i task che compongono i vari servizi, poiché le assunzioni su cui queste sono costruite risultano in molti casi radicalmente diverse da servizio a servizio e difficilmente assimilabili.

L'idea principale attorno alla quale ruota questa proposta è che la soluzione vada ricercata a un altro livello, metodologico, nell'adozione di un approccio interdisciplinare, formale, modulare. Tale approccio consiste nel partire da un'ontologia fondazionale (ovvero una teoria generale che caratterizza proprietà e relazioni fondamentali per ogni tipo di analisi ontologica, a prescindere dall'applicazione specifica) e, a partire da questa, costruire l'ontologia dell'e-Government, specificandone e relativizzando al dominio le proprietà e relazioni. L'attenzione si concentra sull'individuazione di primitive semantiche di base comprensibili per la descrizione dei processi, frutto di un'accurata analisi dei fondamenti del dominio da rappresentare. Non si tratta quindi di costruire un modello standard valido per tutti, quanto piuttosto di creare i presupposti affinché i diversi modelli possano essere prima di tutto compresi e poi integrati. L'ontologia dell'e-Government dovrà comprendere, oltre a un'ontologia dei servizi, anche un'ontologia delle organizzazioni, poiché molto spesso dalla relazione tra le entità di questi due domini che scaturisce la qualità di un servizio.

3.3.2 Obiettivi della proposta

Obiettivo generale di questa proposta quello di incrementare il livello di comunicazione tra i vari attori nei futuri scenari dell'e-Government, grazie all'uso di un approccio ontologico modulare e ben fondato che metta in luce quali sono i mattoni fondamentali sui quali costruire una comunicazione semanticamente adeguata.

L'e-Government consiste nell'utilizzo di tecnologie informatiche nelle procedure amministrative e gestionali della PA. I benefici attesi di questo processo sono molteplici e vanno dalla semplificazione e deburocratizzazione delle procedure amministrative, che diventerebbero di conseguenza pi veloci ed efficienti, all'integrazione dei dati ora distribuiti tra diversi enti, alla riduzione dei costi della PA, alla maggiore trasparenza delle procedure e maggiore partecipazione della cittadinanza.

L'ottimizzazione dei servizi alla persona/cittadino presuppone d'altra parte una condivisione e comunicazione dell'informazione tra enti anche molto eterogenei che rappresentano e gestiscono tale informazione con modalità spesso distinte, con diverse assunzioni riguardo ai termini impiegati e utilizzando strumenti e linguaggi diversi. È quindi fondamentale che il passaggio dell'informazione ne preservi il contenuto e il significato; questa problematica viene spesso etichettata con la locuzione "interoperabilità semantica". L'obiettivo principale della proposta dunque quello di elaborare un modello integrato per l'e-Government fondato su un'ontologia fondazionale di riferimento che costituisca uno strumento idoneo a mettere in relazione le diverse basi di conoscenza di pertinenza dei vari servizi e dei relativi enti eroganti, strutturate secondo schemi e logiche diversi. In altre parole, l'obiettivo non dunque quello di raccogliere la descrizione di tutti gli elementi relativi alla vita della PA per elaborare un unico modello condiviso, quanto quello di analizzare e mettere in comune le primitive di cui queste descrizioni sono costituite, cio le categorie e le relazioni che servono per comprendere e per scambiarsi tali (differenti) descrizioni. Ciò comporterà la costruzione di un modello che integrerà una rappresentazione dei servizi con una rappresentazione dettagliata delle organizzazioni che erogano tali servizi.

Obiettivi specifici:

A. Costruzione di un modello ontologico delle organizzazioni

La Pubblica Amministrazione costituita da enti strutturati, che possono sotto molti rispetti essere visti come casi particolari di organizzazioni. Questo obiettivo specifico consiste nel modellare la struttura degli enti/organizzazioni, ovvero le relazioni che sussistono tra i ruoli che li compongono, le norme che regolano tali relazioni ecc. Astrarre dagli enti specifici e fornire un modello generale di organizzazione importante poiché la Pubblica Amministrazione si trova spesso a interagire con entità collettive di natura diversa, come per esempio imprese private e può addirittura arrivare a dipendere da alcune di esse per l'erogazione di particolari servizi (che possono per esempio essere assegnati in outsourcing). In questi casi di estrema utilità disporre di uno strumento che

permetta di rappresentare enti pubblici e imprese private in uno stesso modello, come specializzazioni di uno stesso genere di entità. La modellazione non si fermerà dunque alla rappresentazione della struttura della PA, ma si estenderà a tutto il contesto sociale e organizzativo nel quale essa è inserita.

B. Costruzione di un modello ontologico dei servizi

I servizi erogati dalla PA sono, oltre che numerosi, molto vari per quanto riguarda le caratteristiche peculiari che li distinguono gli uni dagli altri. Tuttavia, esistono delle proprietà che sono comuni a tutti e che li identificano in quanto servizi (ad esempio di avere un soggetto responsabile per la loro erogazione, una documentazione da produrre ecc.). Questo obiettivo si estrinseca nel rintracciare tali proprietà e rappresentarle in un modello generico, che sia specializzabile in ciascuno dei particolari servizi. È importante non fermarsi alla rappresentazione del workflow, ma rappresentare in maniera rigorosa anche il contenuto dei servizi e, attraverso di esso, la connessione che si instaura tra servizi diversi, in modo da poter facilitare il reperimento di servizi su base semantica.

C. Costruzione di un modello ontologico dell'e-Government risultante dai due punti precedenti

Come negli altri casi di organizzazione, le relazioni tra le entità che compongono gli enti della PA sono spesso regolate da norme e la particolare forma che la struttura dell'ente/organizzazione assume spesso conseguenza della suddivisione di compiti e funzioni che viene effettuata al fine di raggiungere i fini che l'organizzazione si prefigge, che, nel caso degli enti della PA, in generale corrispondono con l'erogazione di servizi.

Questo obiettivo specifico si articola in due sotto-obiettivi: da una parte comprendere che genere di legami sussistano tra i servizi erogati da enti posti in relazione, dall'altra indagare che tipo di relazione si instaura tra enti erogatori di servizi interconnessi. Il primo sott'obiettivo permette di comprendere meglio il funzionamento delle procedure interne alla PA, mentre il secondo facilita l'analisi della distribuzione delle responsabilità. Ciò mostra chiaramente quanto analisi dei servizi e analisi delle organizzazioni siano strettamente complementari (anche se spesso gli studiosi di e-Government si concentrano molto di più sulla prima che sulla seconda) e come solo lo studio congiunto di questi due aspetti possa fornire una spiegazione soddisfacente dell'impatto della tecnologia sulle strutture organizzative, permettendo inoltre di formulare delle previsioni sulla loro evoluzione conseguente a tale impatto. Inoltre, l'applicazione di tale modello a casi reali permette di rintracciare quelle cause di inefficienza che non sono tanto peculiari della sfera tecnologica, quanto di quella sociale.

3.3.3 Stato dell'arte e risultati preliminari

In termini molto generali, in informatica si suole chiamare "ontologia" lo schema concettuale di un certo dominio ([1], [2]). Tale schema concettuale può assumere

forme molto diverse, a partire da una semplice tassonomia fino ad arrivare a una vera e propria teoria logica. Normalmente si tratta di una gerarchia di concetti correlati attraverso relazioni semantiche, ma le ontologie pi elaborate forniscono anche regole (assiomi) che aiutano a specificare come strutturato il dominio. Siamo quindi in presenza di artefatti ingegneristici costituiti da un insieme di termini (vocabolario) relativi a un certo dominio e un insieme di assunzioni esplicite che vincolano l'interpretazione dei termini del vocabolario in modo da avvicinarsi il pi possibile al loro significato inteso. Tale significato viene reso pi esplicito, in maniera tale da ridurre al minimo i fraintendimenti e in ciè consiste il pi importante contributo delle ontologie all'interoperabilit  ([3], [4], [5], [6], [7] e [8]).

Pi in dettaglio, le ontologie fondazionali (come ad esempio CYC, SUMO, DOLCE, BFO, etc.) sono teorie molto generali intese a caratterizzare propriet  e relazioni fondamentali per ogni tipo di analisi ontologica, a prescindere dall'applicazione specifica; esempi di queste propriet  e relazioni sono: identit , parte, dipendenza, partecipazione, costituzione, etc. Queste propriet  e relazioni non sono specifiche di alcun dominio particolare, ma si pu  pensare di esprimere le propriet  e relazioni dei domini specifici come specializzazioni di quelle appartenenti all'ontologia fondazionale. In questo modo, le informazioni di pertinenza di due domini diversi continueranno a essere rappresentate con ontologie diverse, ma avranno ora un "terreno di incontro" nell'ontologia fondazionale. Il Laboratorio di Ontologia Applicata dell'ISTC-CNR ha sviluppato un'ontologia fondazionale (DOLCE: Descriptive Ontology for Linguistic and Cognitive Engineering [9], [10]), largamente utilizzata in numerosi progetti nazionali e internazionali, che si propone di adottare come strumento.

Una volta appurata l'utilit  delle ontologie per l'interoperabilit , risulta abbastanza evidente come l'e-Government si configuri immediatamente come loro terreno di applicazione, come testimoniano i numerosissimi articoli scientifici pubblicati di recente (tra gli altri, si vedano [11], [12] e [13]). L'approccio adottato nella maggioranza dei casi consiste nello sviluppare per ogni servizio di e-Government un'ontologia orientata alle applicazioni, specifica per un task, che sia il pi possibile aderente alla visione degli "esperti" e principali utilizzatori del sistema, ovvero il personale della PA che si occupa del servizio. Le ontologie orientate alle applicazioni hanno lo scopo di caratterizzare un dominio di analisi preciso, individuandone le primitive teoriche pi adatte ad affrontare i problemi per i quali tali ontologie vengono costruite. Affinch  poi queste ontologie siano condivise, vengono creati dei consorzi di PA, all'interno dei quali vengono identificate delle best practices e dei modelli "vincenti" e su questi viene raggiunto un consenso.

Questa soluzione in linea di massima soddisfacente per i casi in cui le informazioni debbano essere condivise tra gli uffici che si occupano di uno stesso servizio all'interno di diverse PA. Tuttavia la situazione si complica nel caso della visione della PA come interlocutore unico per i servizi di e-Government, poich  necessario che l'informazione viaggi anche trasversalmente rispetto ai servizi ed precisamente quest'ultima circostanza a rendere l'applicabilit  dell'approccio del consenso di cui sopra questione particolarmente delicata. Motivo

di ciò che gli stessi dati possono assumere significati anche molto diversi a seconda che si adotti la prospettiva di un servizio piuttosto che di un altro. Si pensi a un esempio molto comune: quando un anziano indigente viene ricoverato in una residenza socio-assistenziale (RSA), lo stesso dato può essere interpretato come data di inizio di una medicalizzazione per il servizio medico-sanitario della RSA e al tempo stesso come data di termine dell'occupazione di un alloggio popolare per il servizio di edilizia abitativa popolare del Comune. A ciò si aggiunga che esiste un altro genere di utente, il cittadino, per il quale non si può presupporre nessuna conoscenza specifica di dominio e che deve ugualmente essere messo in grado di accedere a (almeno parte della) informazione gestita dalla PA. Per maggiore precisione, il problema dell'interoperabilità nella PA può essere articolato secondo diversi livelli e ambiti. I livelli sono:

- integrazione dei dati (tra viste e aspetti diversi della conoscenza sui dati)
- integrazione del workflow (tra servizi e processi diversi)
- integrazione dell'organizzazione (tra servizi e organizzazioni)
- integrazione strategica (tra una PA e il contesto in cui opera). I primi due livelli sono stati oggetto di uno studio molto approfondito nella letteratura relativa alle architetture orientate ai servizi ([14], [15] e [16]), mentre poca attenzione stata finora dedicata agli ultimi due, altrettanto importanti, livelli.

Gli ambiti sono invece i seguenti:

- all'interno di un'unica PA (ad esempio tra uffici che si occupano di servizi diversi)
- tra diverse PA (ad esempio tra gli uffici che si occupano degli stessi servizi)
- tra PA ed entità "esterne" (ad esempio singoli cittadini o imprese private). L'approccio della convergenza sulle ontologie di dominio lascia dunque dei problemi aperti.

3.3.4 Rilevanza della proposta per SPCoop

La proposta consiste nell'ancorare le ontologie specifiche per un dominio a un'ontologia fondazionale o, meglio ancora, costruire le ontologie di dominio come moduli specializzati di una ben fondata ontologia fondazionale. In pratica, si tratta di definire un'ontologia dell'e-Government come modulo dell'ontologia fondazionale DOLCE. Tale ontologia comprenderà un'ontologia dei servizi e una delle organizzazioni, che potranno poi essere istanziate nei particolari servizi e strutture responsabili di tali servizi.

È importante sottolineare l'aspetto innovativo consistente nel rappresentare sia i servizi sia le strutture eroganti all'interno dello stesso modello, poiché ciò permette di avere una pi chiara visione di insieme, a tutto vantaggio della trasparenza amministrativa e gestionale e della maggiore facilità nell'individuare

eventuali inefficienze, le cui cause vanno spesso al di là delle mere disfunzioni a livello di tecnologia. Questo tipo di approccio poco rappresentato in letteratura, in cui i modelli sono centrati quasi esclusivamente sui servizi e le organizzazioni o strutture che erogano i servizi vengono normalmente menzionate tra gli attributi dei servizi stessi e quasi mai esplicitamente rappresentate.

Dal punto di vista della rilevanza strategica, si può notare quanto segue. All'inizio del 2007 il Ministro Nicolais pubblica le linee strategiche per l'e-Government [17]; la sua iniziativa si inserisce in un quadro europeo già abbastanza definito, infatti da alcuni anni, già a partire dal 5^o e 6^o Programma Quadro e dalla cosiddetta strategia di Lisbona, l'Unione Europea ha cominciato a destinare cospicui finanziamenti alla ricerca sull'e-Government. Tuttavia, specialmente nella prima fase di attuazione di queste ricerche, l'attenzione si concentra principalmente sull'ideazione di tecnologie ad hoc, ad esempio tecnologie che facilitassero ai cittadini l'accesso ai dati attraverso form online. La stessa tendenza si è manifestata anche a livello nazionale (non solo in Italia, ma in tutti gli Stati membri dell'Unione, [18]) fino a tempi recenti, privilegiando lo studio e lo sviluppo di tecnologie per l'erogazione di servizi specifici al cittadino.

La conseguenza di tutto ciò è stato un proliferare di diverse prospettive, terminologie, modi di intendere gli stessi oggetti. Un'analisi attenta di questa eterogeneità ha messo in luce l'intrinseca natura interdisciplinare dell'e-Government stesso, che coinvolge aspetti peculiari della scienza dell'informazione, ma anche della scienza economica e della teoria delle organizzazioni in particolare, delle scienze politiche, sociologiche e psicologiche, solo per citarne alcune. Questa ricchezza di contenuti, che di per sé può senz'altro essere percepita come un elemento positivo, d'altro canto, se lo scenario futuro che si auspica per la PA [19] quello che la vede come un interlocutore unico [20], al quale il cittadino può rivolgersi per ottenere una serie di servizi integrati alla persona, rappresenta un fattore di criticità. Questo perché, affinché la PA possa fornire servizi integrati, necessario che si attui l'interoperabilità semantica tra gli enti che la compongono, ovvero che le informazioni possano essere comprese (prima che dalle macchine dagli utenti umani) e trasmesse correttamente (mantenendo il loro significato inteso).

La consapevolezza dell'importanza di questo aspetto va ampliandosi ed è testimoniata a livello europeo dai programmi attivati in tal senso (IDA, IDABC, eTEN)[18]. In Italia, già a partire dal 2002, è stato attivato il progetto PEOPLE, che ha lo scopo di fungere da punto di aggregazione per quei comuni desiderosi di condividere il loro know how e di individuare delle best practices, in direzione di una maggiore omogeneità. Nel 2006 ha inoltre preso avvio anche il progetto interregionale ICAR, teso a incrementare la collaborazione tra enti locali. Tendenzialmente questa ribadita anche dal CNIPA (Centro Nazionale per l'Informatica nella Pubblica Amministrazione), che ha lanciato all'inizio di febbraio 2007 due progetti che condividono questo stesso spirito: ALI, finalizzato a promuovere un nuovo modello di cooperazione per i piccoli Comuni e RIUSO, per il trasferimento - tra diversi contesti organizzativi e funzionali - di una o più applicazioni basate sull'impiego di ICT. Queste azioni del CNIPA seguono la linea instaurata con l'iniziativa rivolta alla creazione di un Sistema Pubblico di Connettività

(SPC) e di un Sistema Pubblico di Cooperazione (SPCoop).

3.3.5 Riferimenti bibliografici

1. Guarino, N., Formal Ontology in Information Systems, in Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998, N. Guarino, Editor. 1998, IOS Press: Amsterdam. p. 3-15.
2. Guarino, N. and M. Musen, Applied Ontology: Focusing on Content. Applied Ontology, 2005(1): p. 1-6.
3. Noy, N.F., Semantic Integration: A Survey Of Ontology-Based Approaches. SIGMOD Record, 2004. 33(4): p. 65-70.
4. Euzenat, J., et al. Ontologies and Semantic Interoperability. in ECAI-02 Workshop on Ontologies and Semantic Interoperability. 2002. Lyon, France: CEUR.
5. Sheth, A. and C. Ramakrishnan, Semantic (Web) Technology In Action: Ontology Driven Information Systems for Search, Integration and Analysis. IEEE Data Engineering Bulletin, 2003(Special issue on Making the Semantic Web Real): p. 40-48.
6. Vetere, G. and M. Lenzerini, Models for semantic interoperability in service-oriented architectures. IBM Systems Journal, 2005. 44(4): p. 887-903.
7. Wache, H., et al. Ontology-Based Integration of Information A Survey of Existing Approaches. in IJCAI01 Workshop: Ontologies and Information Sharing. 2001.
8. Calvanese, D., G. De Giacomo, and M. Lenzerini. Ontology of integration and integration of ontology. in Description Logic Workshop. 2001.
9. Gangemi, A., et al., Sweetening Ontologies with DOLCE, in Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, A. Gomez Perez and V.R. Benjamins, Editors. 2002, Springer Verlag: Berlin. p. 166-181.
10. Masolo, C., et al., WonderWeb Deliverable D18. 2003.
11. Abecker, A., et al., Semantic Web Meets eGovernment Papers from 2006 AAAI Spring Symposium. 2006, AAAI.
12. Barone, A. and P. Di Pietro, Semantic of eGovernment Processes: a Formal Approach to Service Definition. 2005, Diviana Consulting.
13. Corradini, F., A. Polzonetti, and R. Pruno. E-government administrative and semantic cooperation: the intelligent document role. in EGOV2005. 2005. Linz.
14. Janssen, M. and R. Wagenaar, From Legacy to Modularity: A Roadmap Towards Modular Architectures Using Wb Services Technology, in Electronic Government. 2003, Springer: Berlin/Heidelberg. p. 95-100.
15. Mandell, D. and S. McIlraith. Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation. in 2nd International

- Semantic Web Conference (ISWC03). 2003.
16. Traverso, P. and M. Pistore. Automatic Composition of Semantic Web Services into Executable Processes. in International Semantic Web Conference (ISWC). 2004.
 17. Verso il sistema nazionale di e-government - Linee strategiche. 2007, Ministro per le Riforme e le Innovazioni nella Pubblica Amministrazione: Roma.
 18. Bicking, M. and M.A. Wimmer. eGovernment Research in Europe: Disciplinary Understanding and State of Play from eGovRTD2020. in EGOV2006. 2006. Linz.
 19. Project n. 27139 Roadmapping eGovernment RTD 2020: Visions and Research. Measures towards Citizenship and Innovative Government. Deliverable D2.1: Scenarios report (including regional workshops report). 2006, University of Koblenz-Landau.
 20. Wimmer, M.A. and E. Tambouris. Online One-Stop Government: A working framework and requirements. in IFIP World Computer Congress. 2002. Montreal: Kluwer Academic Publishers.

3.4 Un'ontologia di riferimento per la PA

NOTA: questa sezione è ancora da completare.

Basato su materiale contribuito da Aldo Gangemi
CNR-ISTC e CNIPA

3.4.1 Introduzione

Sulla scorta dei metodi presentati nelle sezioni precedenti e prima di presentare metodi orientati specificamente all'implementazione di servizi semantici, presentiamo qui una proposta di ontologia di riferimento per la PA, basata su alcuni design pattern estratti dall'ontologia fondazionale DOLCE [1] e dalle sue estensioni per la rappresentazione di processi, servizi, organizzazioni, ruoli, oggetti informativi, etc. ([2]).

Sono state inclusi tipi di entità ricorrenti nella descrizione di servizi in SPCoop, sulla base delle esperienze documentate in letteratura [3].

Si presentano infine mappature a OWL-S [4], WSMO [5], SAWSDL-S [6], a ontologie per l'e-Gov e a ontologie attualmente molto usate in ambito Web2.0 e in applicazioni leggere di Web Semantico (FOAF [7], DC [8], etc.). Un obiettivo previsto è l'allineamento con l'ontologia di riferimento per la PA nella EU che il progetto SemanticGov [9] sta attualmente sviluppando (ma ancora nessun draft pubblico è stato pubblicato).

3.4.2 Architettura modulare

L'ontologia di riferimento per la PA, chiamata provvisoriamente "ORPA", è stata progettata riusando alcuni *content design patterns* estratti da DOLCE+, una libreria di ontologie di riferimento, usate in vari progetti italiani e internazionali. Una versione OWL delle ontologie di DOLCE+ si può scaricare da: <http://wiki.loa-cnr.it/index.php/LoaWiki:Ontologies>. L'ontologia di riferimento per la PA, attualmente in evoluzione, si può scaricare da: <http://www.loa-cnr.it/ontologies/CNIPA/orpa.owl>.

L'architettura generale di ORPA e delle ontologie riusate presentata in Fig. 3.8.

ORPA importa un'ontologia dei modelli di piani e processi (PlansLite.owl), un'ontologia degli oggetti informativi (IOLite.owl), un'ontologia dei sistemi (SystemsLite.owl) e un'ontologia del dominio legale (CoreLegal.owl). Tutte queste ontologie importano a loro volta il kernel di DOLCE+, cioè DOLCE+ Ultralite (DUL.owl).

Le ontologie importate non vanno viste come blocchi interamente necessari e prescrittivi rispetto a ORPA, ma piuttosto come un insieme di *content design patterns* utili a sistematizzare i tipi di concetti e relazioni necessari a produrre e mappare le ontologie per le applicazioni della PA.

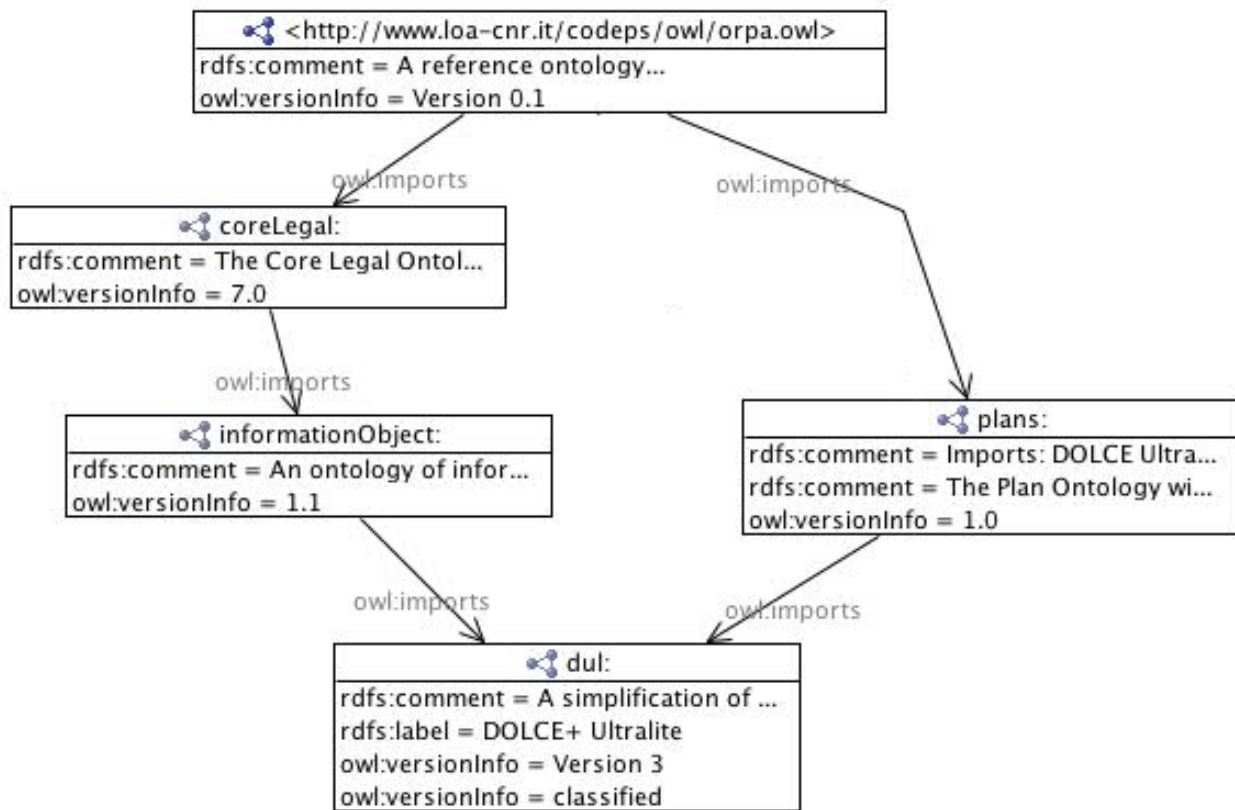


Figura 3.8: L'architettura modulare dell'ontologia di riferimento per la PA

3.4.3 Struttura interna

Riportiamo qui alcuni diagrammi di ORPA, generati in un profilo UML per OWL. I diagrammi presentano gli elementi dell'ontologia in inglese, ma ORPA ha anche una lessicalizzazione italiana, come anche le ontologie che importa.

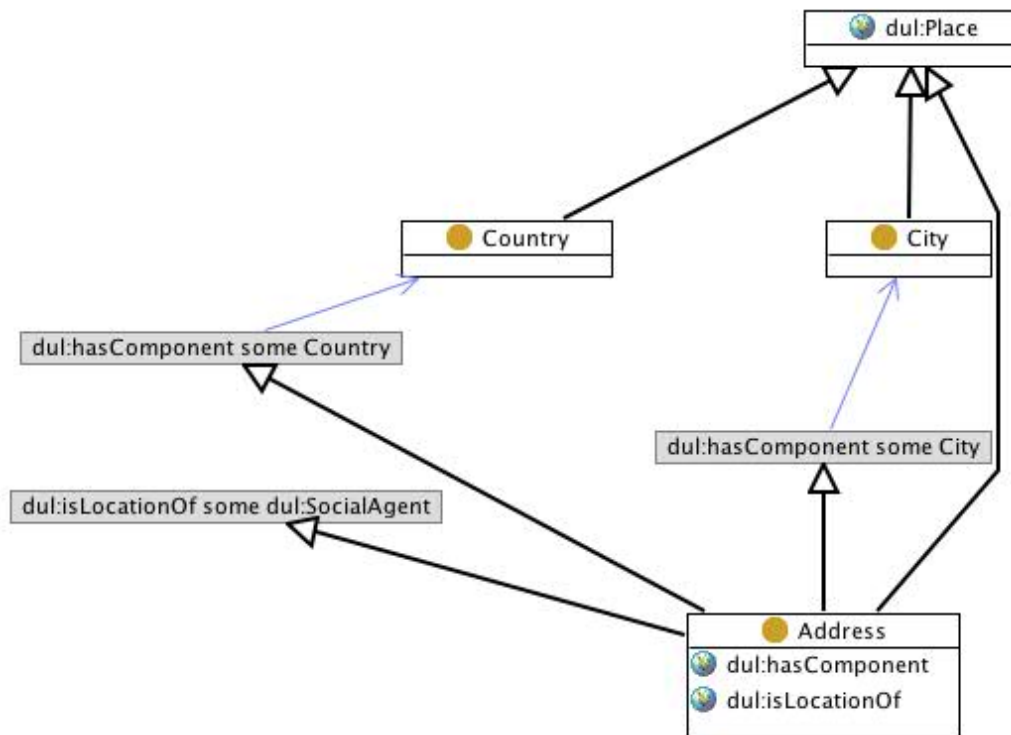


Figura 3.9: La classe “indirizzo” in ORPA. Riusa il pattern “location in place”

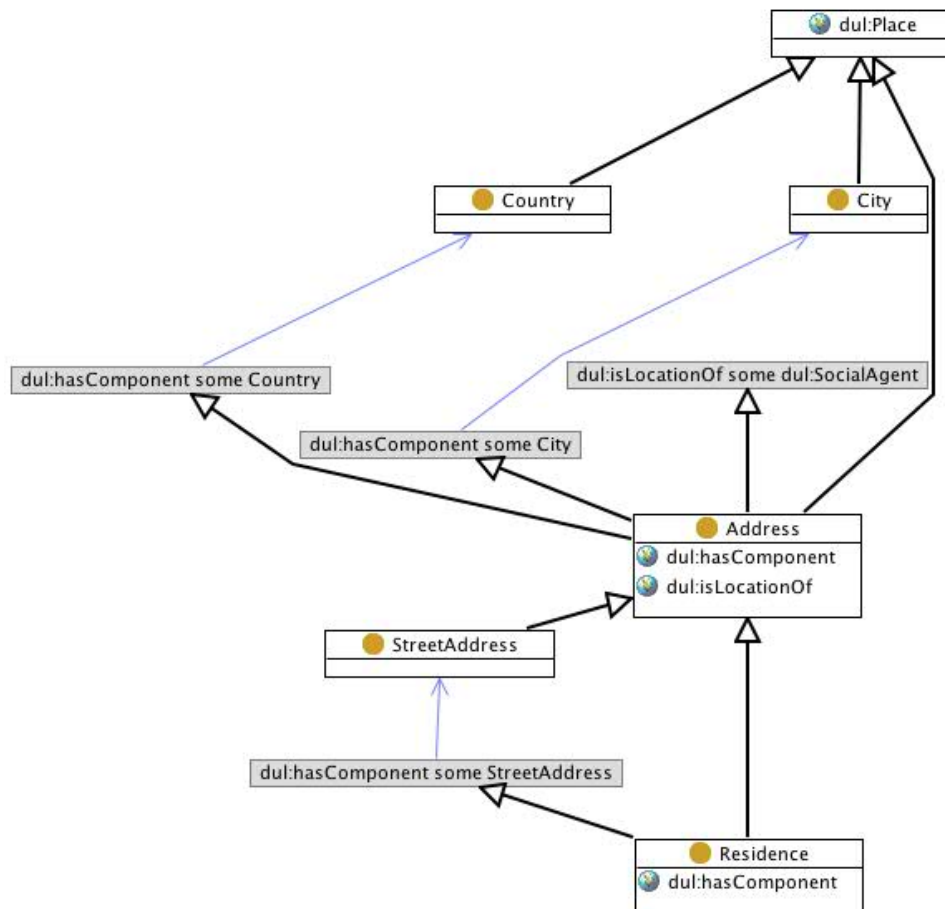


Figura 3.10: La classe “residenza” in ORPA. Riutilizza il pattern “componency”

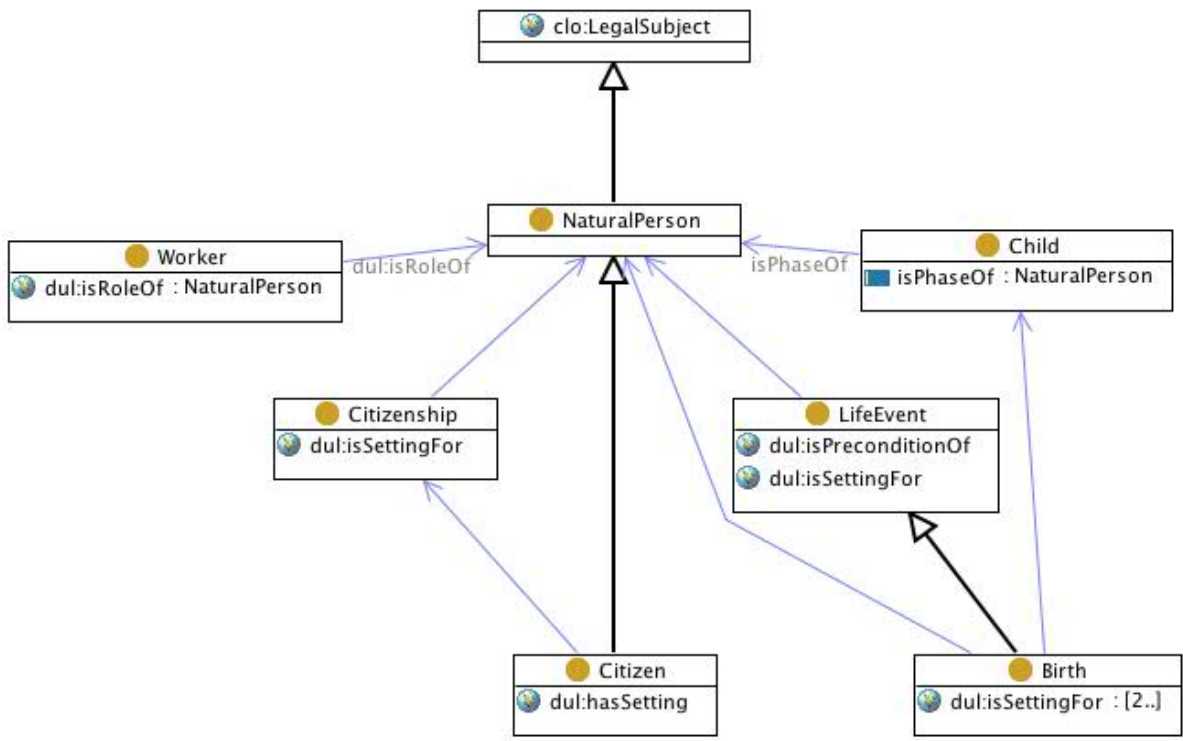


Figura 3.11: La classe “persona naturale” in ORPA. Riusa i pattern “phases”, “agent-role”

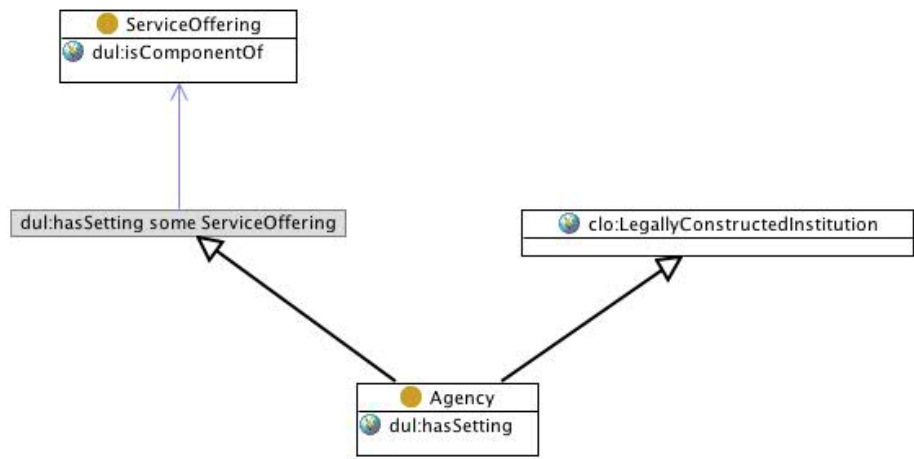


Figura 3.12: La classe “istituzione pubblica” in ORPA

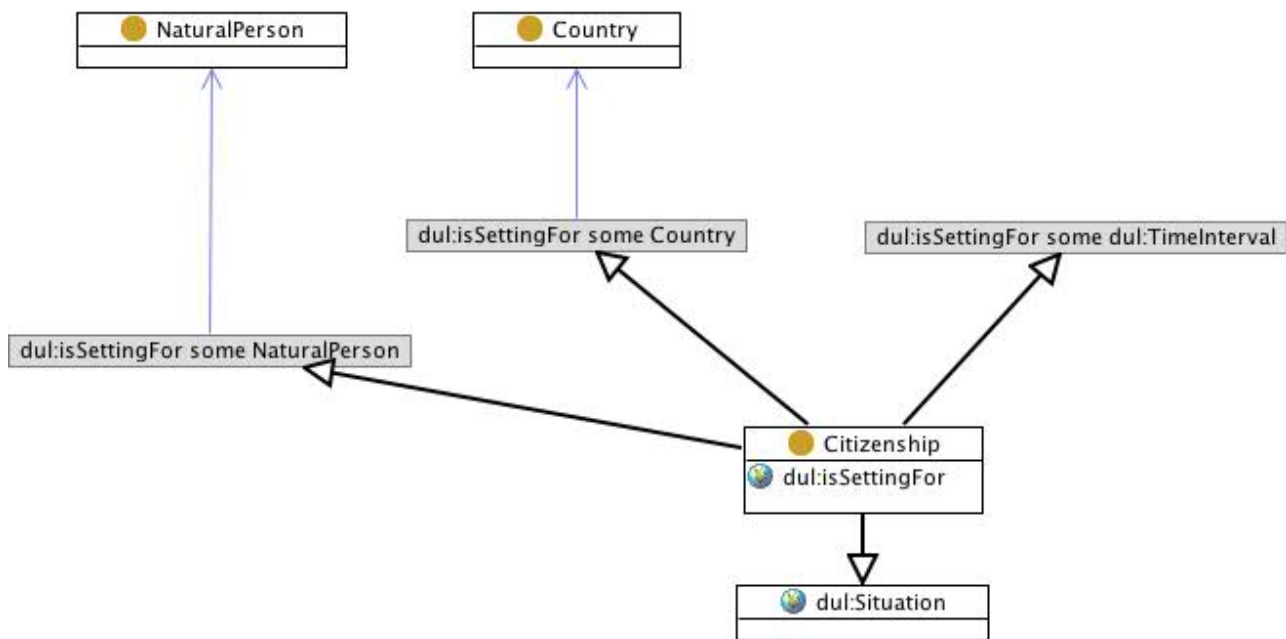


Figura 3.13: La classe “cittadinanza” in ORPA. Riuse il pattern “situation”

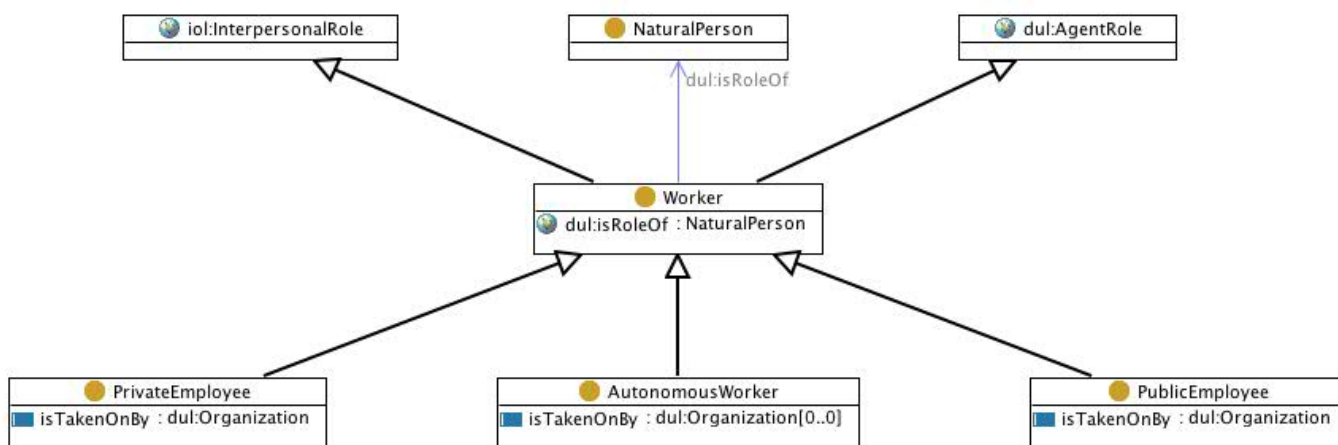


Figura 3.14: La classe “lavoratore” in ORPA. Riuse il pattern “agent-role”

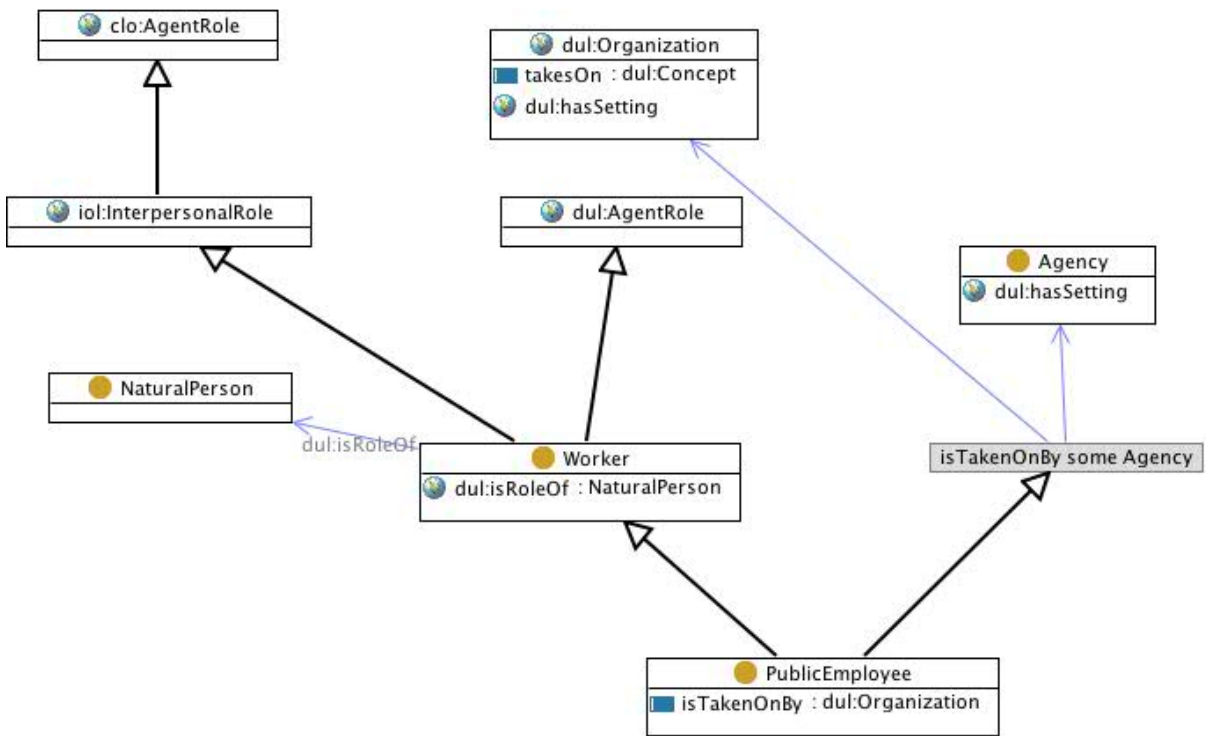


Figura 3.15: La classe “dipendente pubblico” in ORPA. Riusa il pattern “takingOn”

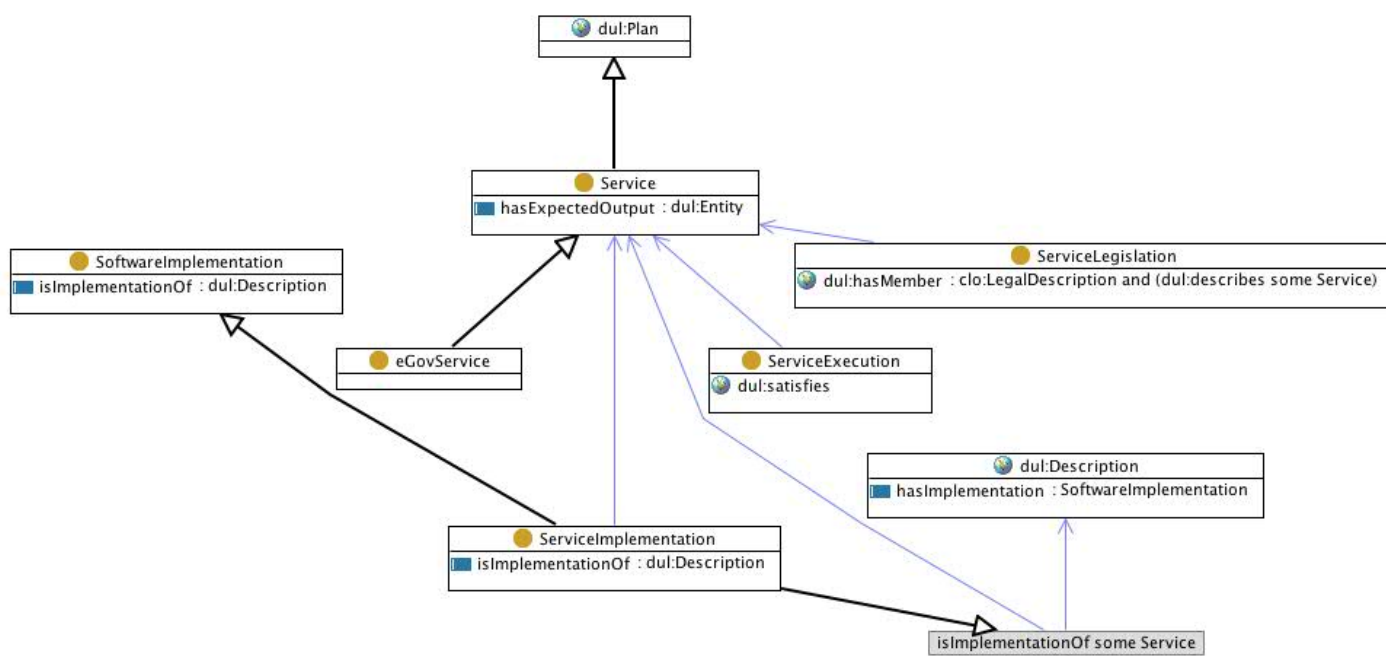


Figura 3.16: La classe “servizio” in ORPA. Riutilizza il pattern “plan”

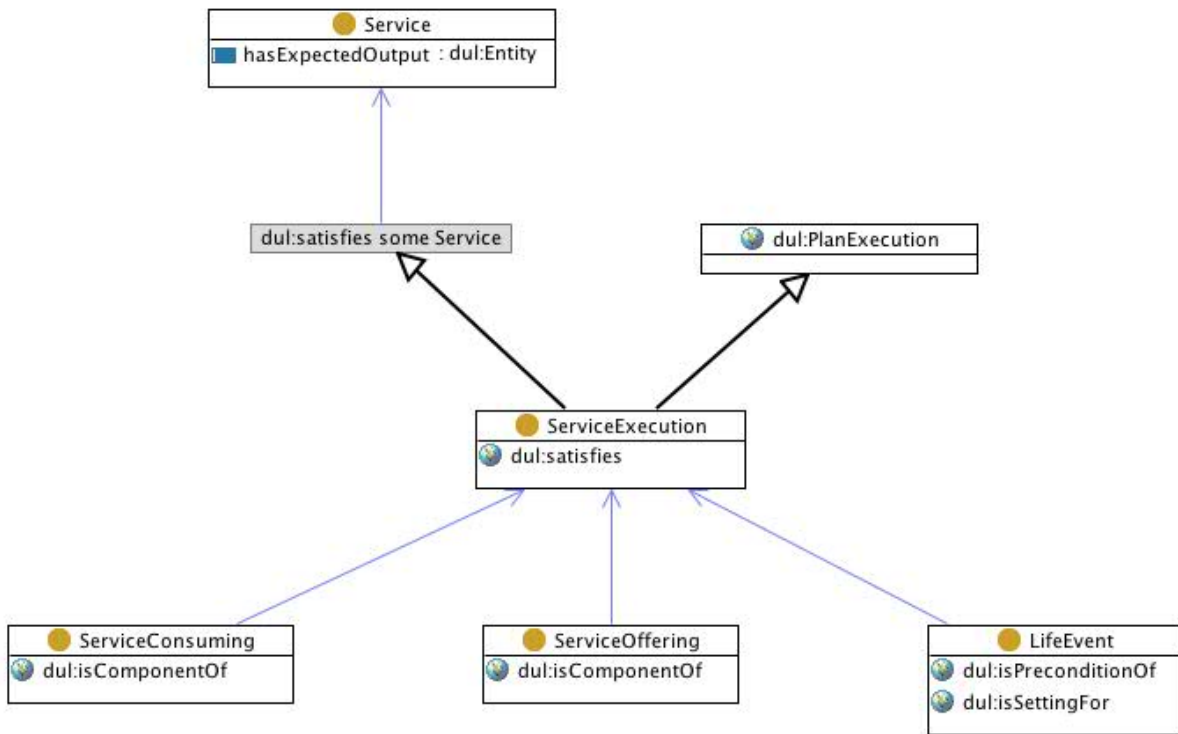


Figura 3.17: La classe “esecuzione di servizio” in ORPA. Riusa il pattern “plan-execution”

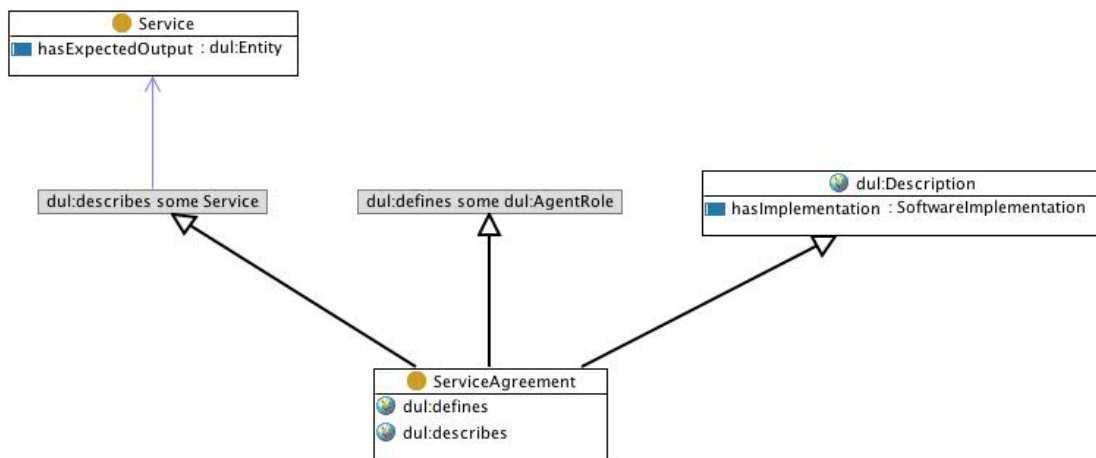


Figura 3.18: La classe “accordo di servizio” in ORPA. Riusa il pattern “description”

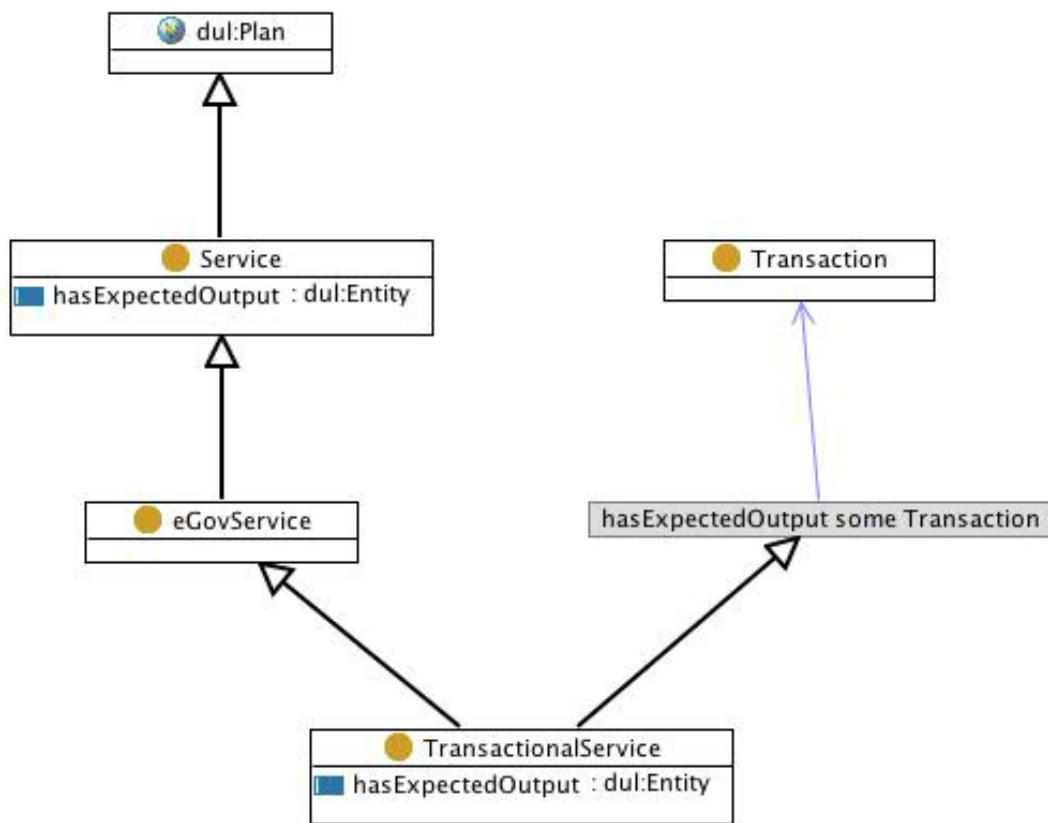


Figura 3.19: La classe “servizio transazionale” in ORPA. Riutilizza il pattern “output”

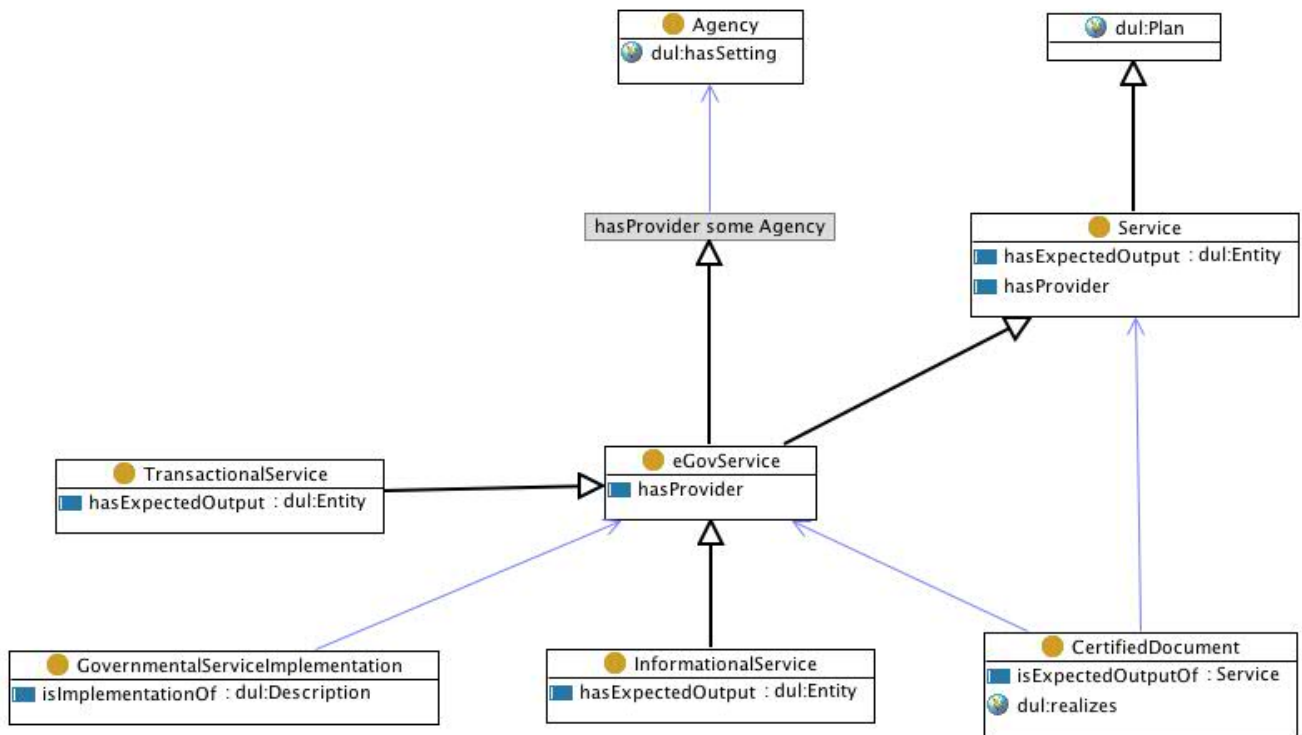


Figura 3.20: La classe “servizio di e-Gov” in ORPA. Riutilizza il pattern “plan” e “provision”

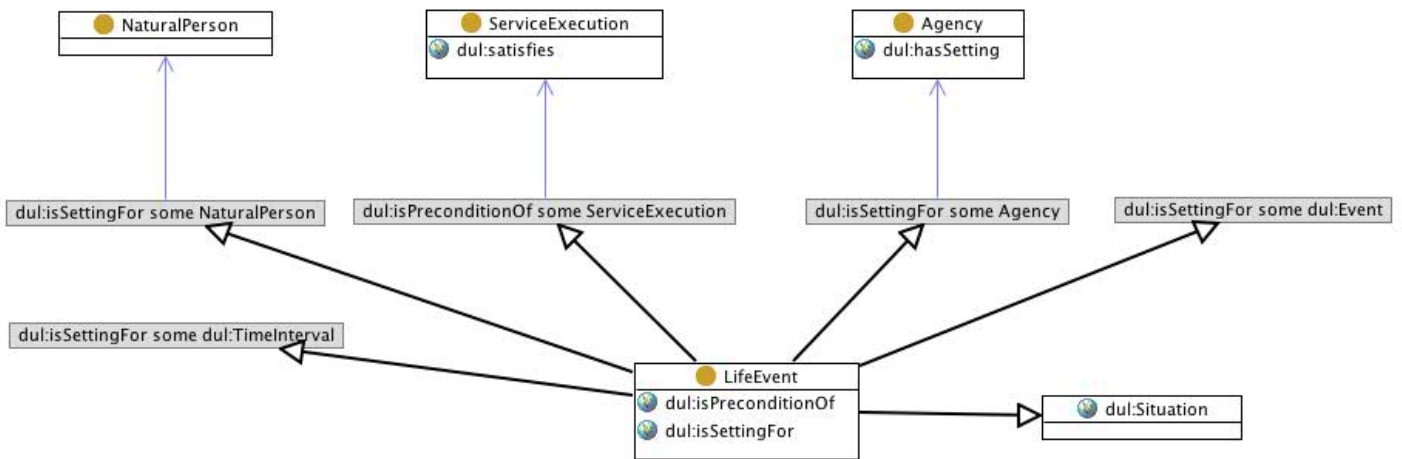


Figura 3.21: La classe “evento della vita” in ORPA. Riutilizza il pattern “situation”

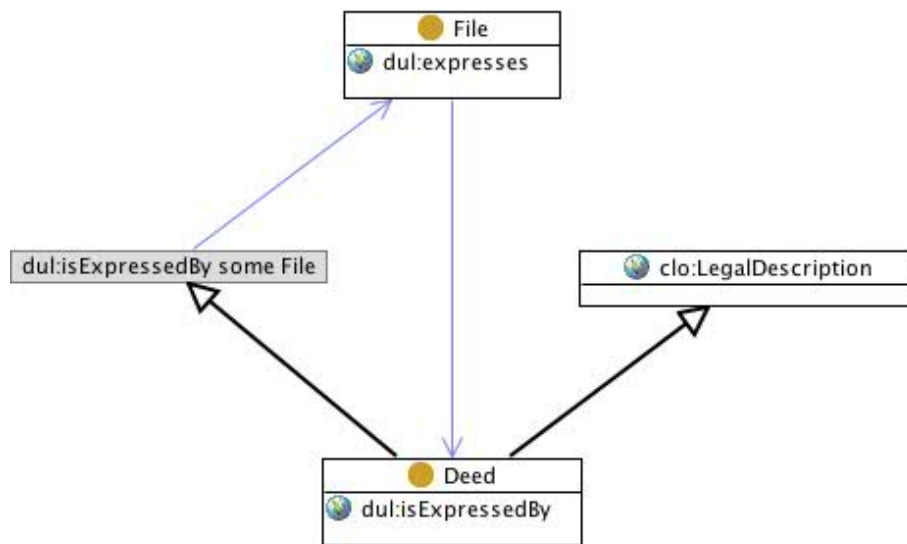


Figura 3.22: La classe “atto” in ORPA. Riusa il pattern “expression”

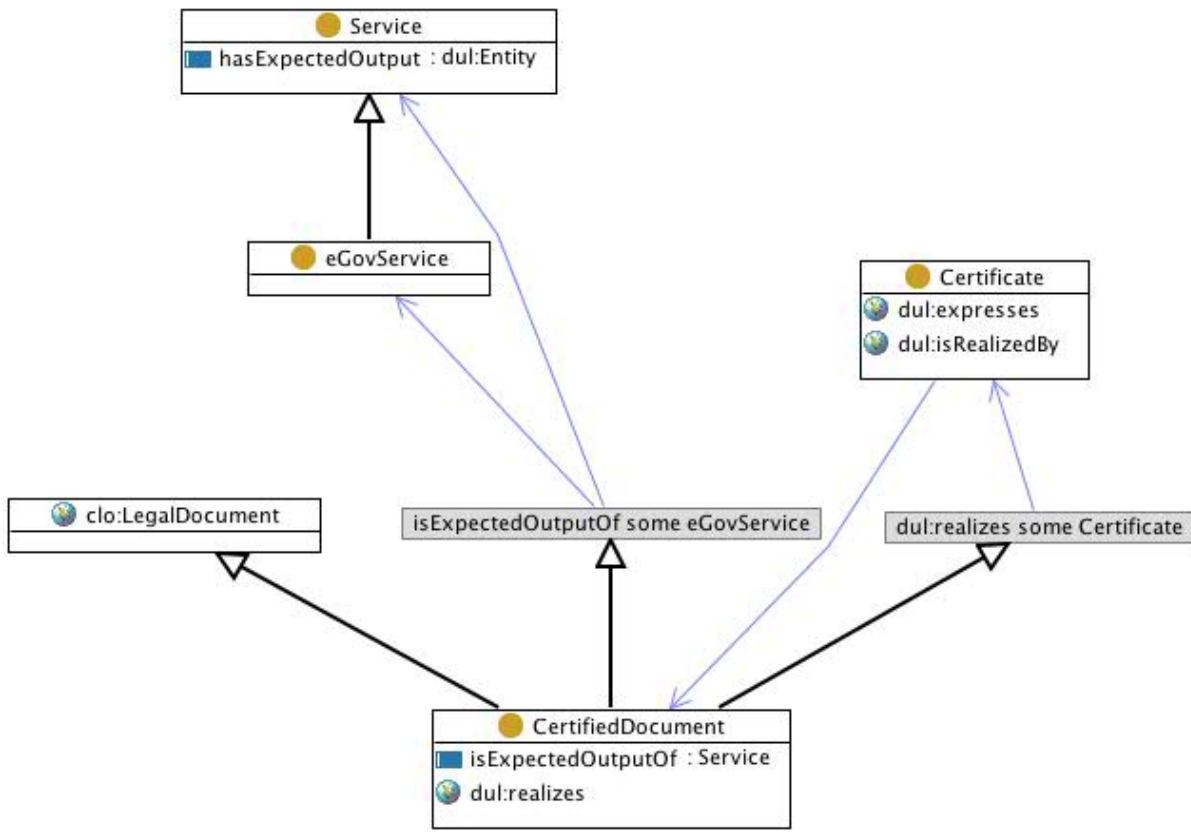


Figura 3.23: La classe “certificato” in ORPA. Riutilizza il pattern “realization”

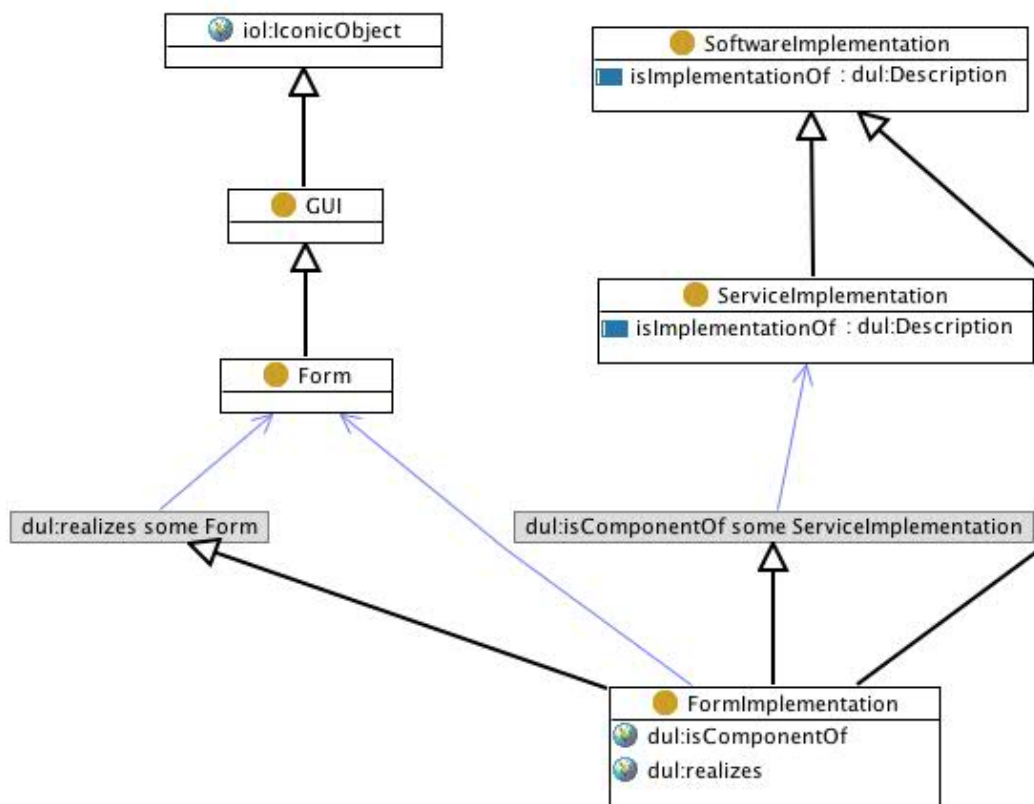


Figura 3.24: La classe “modulo” in ORPA. Riuser i pattern “realization” e “componency”

3.5 SAWSDL e la semantica dei servizi

Abbiamo introdotto metodi e modelli riusabili per le ontologie nella PA. Abbiamo anche introdotto alcuni modelli per rappresentare profili e modelli di processo rispetto a servizi di e-Gov. Rimane però da spiegare come tutta questa semantica si possa raccordare e diventare utile non solo per navigare fra le entità e le informazioni che i servizi della PA gestiscono, ma anche per annotare ed eventualmente meglio articolare i servizi stessi. A questo scopo, diversi approcci sono stati presentati in letteratura, ma i progettisti di servizi sono in gran parte rimasti fedeli al protocollo WSDL, che solo con artifici complessi poteva essere collegato a ontologie per i servizi, come OWL-S e WSMO. Il W3C, con SAWSDL-S, ha permesso di aggiungere semplicemente alcuni attributi a WSDL, a cui si possano agganziare annotazioni semantiche rigorose. Ma SAWSDL-S di per s non rappresenta la soluzione. Come descrivono per esempio (Martin, Paolucci, Wagner, 2007), “SAWSDL represents a conservative, incremental approach to introducing semantic characterization of Web services into mainstream Web service practices ... For example, it aims to provide semantic characterization of a services input and output types, which can be useful in disambiguating those types in the context of simple forms of service discovery. But it does not attempt to provide a comprehensive framework to support more sophisticated approaches to discovery, composition, or any of the other service-related tasks that Semantic Web services research aims to automate. SAWSDL does not specify a particular semantic framework within which to characterize the semantics of Web services. Rather, it defines a small set of WSDL extension attributes, which may be used to refer to constructs within any external semantic framework. SAWSDL is completely noncommittal regarding the choice of semantic framework. It is important to understand that SAWSDL is of very little use unless there is an additional specification of conventions and guidelines for what can be referred to in a particular semantic framework, and what it means to do so. Consequently, such a specification is an essential and timely next step in bringing Semantic Web services research to fruition.”

Nella prossima sezione vediamo come si possa riprendere il discorso dall'interoperabilità alle ontologie ai protocolli di servizio e offrire comunque una piattaforma agile e flessibile, in cui fare annotazioni leggere, oppure usare ontologie articolate come quelle descritte nelle sezioni precedenti.

3.6 Annotazione semantica di Servizi Web

Basato su materiale contribuito da: Marco Pistore e Michele Trainotti
Università di Trento

3.6.1 Introduzione

Il requisito di interoperabilità e cooperazione tra i diversi domini informativi rappresenta una delle sfide più importanti lanciate alla società dell'informazione.

L'interoperabilità facilita la comunicazione, l'interazione e le transazioni tra strutture o partner diversi. A livello informatico l'interoperabilità è definita come la capacità di due o più sistemi informativi di scambiarsi informazioni e di attivare processi elaborativi nei rispettivi domini applicativi.

Nel panorama attuale dove le relazioni tra le varie organizzazioni ed istituzioni sono così fluide è necessario adottare un modello di cooperazione ed interoperabilità che consenta di conservare l'indipendenza dei domini informativi pur garantendo lo scambiando di informazioni. In questo senso l'utilizzo del paradigma basato sui Servizi Web risulta essere la scelta ottimale.

La soluzione dei problemi legati all'aspetto tecnologico non è di per se sufficiente per garantire la completa interoperabilità tra i sistemi in modo automatico. Uno degli aspetti fondamentali ad una forma di interoperabilità più fattiva è la componente semantica. La semantica permette di arricchire l'aspetto legato alle informazioni scambiate tra i sistemi permettendo contestualizzarle rispetto al dominio informativo interno. In sostanza permette da corpo ai singoli elementi informativi consentendo di capire in modo automatico il significato delle informazioni scambiate.

Uno dei domini maggiormente attivi nell'ambito della definizione di norme e standard che puntano a raggiungere una piena interoperabilità è sicuramente quello dell'eGovernment. In questo contesto infatti i progetti attuali si concentrano sullo sviluppo della dimensione di cooperazione e collaborazione tra sistemi delle varie amministrazioni.

L'obiettivo del documento è proporre una rassegna delle tecniche e dei linguaggi per l'annotazione semantica di Servizi Web che ad oggi possono essere utilizzati nell'ambito della cooperazione tra diversi domini informativi.

Questa sezione è strutturata in tre sotto-sezioni. La sotto-sezione *Interoperabilità e Servizi Web* si chiariscono il ruolo dei *Servizi Web* e della *semantica* nel contesto dell'interoperabilità. La sotto-sezione *Semantica e Servizi Web* propone una panoramica delle tecniche e dei linguaggi di modellazione semantica nell'ambito dei Servizi Web che ad oggi sembrano più promettenti. Nella sotto-sezione sono trattati OWL/OWLS, WSMO/WSML, WSDLs e SAWSDL. Infine nella sotto-sezione *Conclusioni* si fa il punto rispetto alle tecniche e ai linguaggi presentati nel documento.

3.6.2 Interoperabilità e Servizi Web

Nel quadro dell'eGovernment l'interoperabilità garantisce in modo automatico ai sistemi informativi l'accesso e la fruizione alle varie risorse informative dei vari enti locali (PAL) e centrali (PAC) presenti sul territorio che ad oggi richiedono la mediazione degli operatori umani.

Allo stato attuale la maggioranza dei sistemi informativi è stata sviluppata considerando l'operatore come fruitore primario dei servizi forniti. E' chiaro che in questo contesto l'esigenza di predisporre un modello di presentazione delle informazioni utilizzabile da applicativi non era rilevante. Come abbiamo visto nel panorama odierno la sfida dell'interoperabilità impone un cambio di paradigma.

Progettare sistemi che consentano lo scambio di informazioni e più in generale supportino l'interoperabilità applicativa impone la necessità di accordarsi su una varietà di elementi complessi spesso dipendenti dal dominio organizzativo dove l'informazione viene creata ed utilizzata.

La semantica è lo strumento che garantisce la possibilità che sistemi informativi distribuiti, basati su differenti concetti associati alle informazioni scambiate, possano collaborare in modo automatico anche se sono stati progettati in modo indipendente. In sintesi l'obiettivo è di fare in modo che le fonti informative siano in rete e che l'informazione scambiata sia automaticamente comprensibile e utilizzabile da applicazioni non coinvolte nella loro creazione.

La European Interoperability Framework distingue tre livelli interoperabilità: *livello tecnico*, *livello semantico* e *livello organizzativo*:

1. L'interoperabilità a *livello tecnico* consente ai sistemi informativi delle organizzazioni di scambiarsi dati e servizi, attraverso interfacce, formati e protocolli standard.
2. L'interoperabilità a *livello semantico* consente di attribuire il medesimo significato alle informazioni scambiate automaticamente fra i sistemi informativi cooperanti.
3. L'interoperabilità a *livello organizzativo* consente di attribuire ruoli, doveri e responsabilità precise ad ogni organizzazione coinvolta in processi condivisi.

Interoperabilità tecnica

L'interoperabilità a livello tecnico vede nella definizione di norme, di specifiche, e di interfacce aperte l'elemento abilitante alla cooperazione tra i sistemi informativi dei vari domini amministrativi.

Questa problematica è stata indirizzata in modo efficace dalle specifiche SPCoop emanate dal Centro Nazionale per l'Informatica nella Pubblica Amministrazione (CNIPA). SPCoop identifica nell'architettura orientata ai servizi (SOA) nella specifica declinazione dei Web Services l'elemento abilitante alla interoperabilità a livello tecnico.

Il paradigma di integrazione basato su *servizi web* impone di considerare il sistema informativo aziendale come erogatore e fruitore di servizi. In ambito SOA un *servizio* rappresenta un'interfaccia verso il sistema informativo interno che descrive un modello dati ed una serie di operazioni consistenti che agiscono su di esso.

L'architettura orientata ai servizi trova nei *Web Service* una sua realizzazione diretta e pratica. Uno dei fattori caratterizzanti dei Web Service è il fatto di essere basati su linguaggi e standard aperti. In particolare l'interfaccia di un Web Service viene descritta in WSDL (Web Service Description Language) mentre i metodi sono attivabili tramite lo scambio di messaggi SOAP (Simple Object Access Protocol) che, solitamente, vengono trasportati tramite il protocollo HTTP.

Interoperabilità semantica

L'interoperabilità a livello semantico trova nella definizione di ontologie e di efficienti tecniche di matching lo strumento principale per la sua realizzazione. Allo stato attuale esistono diverse tecniche e linguaggi utilizzabili in ambito semantico. Dal punto di vista dei linguaggi i lavori più interessanti sono quelli relativi a OWLS, WSMO, WSDLS e SAWSDL. Nel quadro di riferimento una delle tecniche che sembra essere interessante è l'arricchimento dei servizi con informazioni semantiche.

Interoperabilità organizzativa

L'interoperabilità a livello organizzativo si riferisce alla definizione e descrizione formale dei processi e delle procedure organizzativo/amministrative interente. E' importante sottolineare che l'interoperabilità organizzativa non può essere attuata in modo automatico prescindendo dai livelli di interoperabilità tecnica e di interoperabilità semantica. In particolare la certezza della compatibilità ontologica dei concetti ai quali le informazioni scambiate si riferiscono è essenziale.

3.6.3 Semantica e Servizi Web

Lo strumento per contestualizzare in modo semantico le applicazioni ed i servizi di un sistema informativo è l'ontologia. L'ontologia permette di descrivere le entità di un dominio e le relazioni tra esse. In pratica un'ontologia identifica e definisce un vocabolario del dominio applicativo di riferimento.

In generale sistemi diversi sono basati su ontologie di riferimento diverse. Lo sforzo di integrazione si concentra quindi nella derivazione di una ontologia comune tra i sistemi che permetta la comunicazione e la cooperazione tra essi.

Nel caso del Web, le ontologie descrivono esattamente la descrizione delle informazioni e delle relazioni tra le informazioni e aprono la strada al Semantic Web. Annotando semanticamente dati ed informazioni nello spazio Web è possibile delegare alla macchina un ruolo primario nelle fasi di ricerca, selezione, utilizzo, composizione e monitoring dei servizi Web.

Nel campo delle ontologie ci sono ad oggi diversi linguaggi che hanno raggiunto un certa maturità: OWL/OWLS, WSML/WSMO, WSDLS e SAWSDL.

Nel seguito di questa sezione viene data una panoramica dei vari linguaggi applicate ad uno scenario di riferimento.

Lo scenario di riferimento individuato è quello di un servizio in ambito sanitario che restituisce l'anagrafica sanitaria di un assistito. Il servizio mette a disposizione un unico metodo *IdentificaAssistito* che si aspetta di ricevere il *nome*, il *cognome* e la *data di nascita* dell'assistito da ricercare e restituisce l'*anagrafica dell'assistito*, la *lista delle esenzioni* cui lo stesso ha diritto ed infine l'*anagrafica del medico curante*. In Figura 3.25 riportiamo un estratto del WSDL del servizio dello scenario di riferimento.

L'ontologia dello scenario di riferimento è molto semplice e definisce i seguenti concetti:

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://it.tn.apss/VisuraDatiSanitari/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name="interfaociaImplementativaErogatore"
  targetNamespace="http://it.tn.apss/VisuraDatiSanitari/">

  <wsdl:types>
    <xsd:schema targetNamespace="http://it.tn.apss/VisuraDatiSanitari/"
      xmlns:tns="http://it.tn.apss/VisuraDatiSanitari/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
      ...
      <xsd:complexType name="AnagrafiaAssistito">
        <xsd:sequence>
          <xsd:element name="oodSanitario" type="tns:CodiceSanitarioNazionale"/>
          <xsd:element name="oodFiscale" type="tns:CodiceFiscale"/>
          <xsd:element name="oognome" type="tns:Cognome"/>
          <xsd:element name="nome" type="tns:Nome"/>
          <xsd:element name="sesso" type="tns:Genere"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="AnagrafiaMedioDiBase">
        <xsd:sequence>
          <xsd:element name="oodFiscale" type="tns:CodiceFiscale"/>
          <xsd:element name="oognome" type="tns:Cognome"/>
          <xsd:element name="nome" type="tns:Nome"/>
          <xsd:element name="sesso" type="tns:Genere"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="Esenzione">
        <xsd:sequence>
          <xsd:element name="numeroEsenzione" type="xsd:string"/>
          <xsd:element name="oodPatologia" type="xsd:string"/>
          <xsd:element name="dataInizioEsenzione" type="xsd:date"/>
          <xsd:element name="dataScadenzaEsenzione" type="xsd:date" minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="Esenzioni">
        <xsd:sequence>
          <xsd:element name="esenzione" type="tns:Esenzione"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:schema>
  </wsdl:types>

  <wsdl:message name="richiesta_RichiestaRispostaSinorona_IdentifioaAssistito_Msg">
    <wsdl:part name="oognome" type="tns:Cognome"/>
    <wsdl:part name="nome" type="tns:Nome"/>
    <wsdl:part name="dataNasoita" type="xsd:date"/>
  </wsdl:message>

  <wsdl:message name="risposta_RichiestaRispostaSinorona_IdentifioaAssistito_Msg">
    <wsdl:part name="anagrafiaAssistito" type="tns:AnagrafiaAssistito"/>
    <wsdl:part name="medioDiBase" type="tns:AnagrafiaMedioDiBase"/>
    <wsdl:part name="listaEsenzione" type="tns:Esenzioni"/>
  </wsdl:message>

  <wsdl:message name="fault_RichiestaRispostaSinorona_IdentifioaAssistito_Msg">
    <wsdl:part name="oognome" type="tns:Cognome"/>
    <wsdl:part name="nome" type="tns:Nome"/>
    <wsdl:part name="dataNasoita" type="xsd:date"/>
    <wsdl:part name="errore" type="xsd:integer"/>
    <wsdl:part name="erroreDeso" type="xsd:string"/>
  </wsdl:message>

  <wsdl:portType name="VisuraDatiSanitari_PT">
    <wsdl:operation name="IdentifioaAssistito">
      <wsdl:input
        message="tns:richiesta_RichiestaRispostaSinorona_IdentifioaAssistito_Msg" />
      <wsdl:output
        message="tns:risposta_RichiestaRispostaSinorona_IdentifioaAssistito_Msg" />
      <wsdl:fault name="ErroreApploativo" />
      <message="tns:fault_RichiestaRispostaSinorona_IdentifioaAssistito_Msg"/>
    </wsdl:operation>
  </wsdl:portType>
  ...
</wsdl:definitions>

```

Figura 3.25: WSDL del servizio dello scenario di riferimento

- Nome
- Cognome
- DataDiNascita
- CodiceFiscale
- CodiceSanitarioNazionale
- Persona
- Assistito
- Medico
- MedicoDiBase

OWL

Web Ontology Language (OWL) è un linguaggio per definire ontologie web basato sui precedenti OIL e DAML+OIL. Il riconoscimento di OWL come W3C recommendation ha portato allo sviluppo di un crescente insieme di tools e tecniche di ricerca che come obiettivo l'introduzione della semantica negli ambienti di produzione.

OWL è stato progettato e sviluppato con l'intento di unificare il modo di processare il contenuto semantico e l'informazione web; è basato su XML e questo ne facilita l'uso nell'abito di sistemi eterogenei e distribuiti.

In Figura 3.26 è riportato uno stralcio del codice OWL dell'ontologia definita per lo scenario di riferimento.

```

...
<owl:Class rdf:ID="Assistito">
  <rdf:subClassOf>
    <owl:Class rdf:ID="Persona"/>
  </rdf:subClassOf>
</owl:Class>
<owl:Class rdf:ID="DataDiNascita"/>
<owl:Class rdf:ID="Cognome"/>
<owl:Class rdf:ID="Medico">
  <rdf:subClassOf rdf:resource="#Persona"/>
</owl:Class>
<owl:Class rdf:ID="ListaDiEsenzioni"/>
<owl:Class rdf:ID="CodiceSanitarioNazionale"/>
<owl:Class rdf:ID="CodiceFiscale"/>
<owl:Class rdf:ID="Esenzione"/>
<owl:Class rdf:ID="MedicoDiBase">
  <rdf:subClassOf rdf:resource="#Medico"/>
</owl:Class>
<owl:Class rdf:ID="Nome"/>
...

```

Figura 3.26: OWL per l'ontologia dello scenario di riferimento

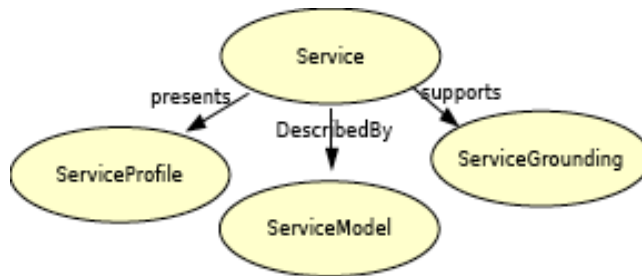


Figura 3.27: Concetti definiti in OWLS

OWLS

OWLS è un'ontologia specificatamente pensata per l'ambito dei Web Service basata su OWL. In sintesi permette di descrivere da una parte l'ontologia del dominio di riferimento e dall'altra permette di descrivere in modo efficace *cosa fa un servizio web, come deve essere usato e come si interagisce con esso*.

Come specificato in Figura 3.27, OWLS organizza la descrizione di un servizio nei quattro concetti *Service*, *Service Profile*, *Service Model* e *Service Grounding*.

Service identifica un servizio e permette di legare tra loro tramite le proprietà *presents*, *describedBy* e *supports* rispettivamente il *Service Profile*, il *Process Model* ed il *Grounding* di uno specifico servizio. In Figura 3.28 è proposto un stralcio della definizione del *Service* del servizio di riferimento per lo scenario selezionato.

```

...
<service:Service rdf:ID="IdentifioaAssistitoService">
  <service:supports rdf:resource="#IdentifioaAssistitoAtomioProoessGrounding"/>
  <service:describedBy rdf:resource="#IdentifioaAssistitoProoess"/>
  <service:presents rdf:ID="IdentifioaAssistitoProfile"/>
</service:Service>
...

```

Figura 3.28: OWLS del servizio dello scenario di riferimento

Il *Service Profile* descrive le funzionalità e le proprietà non funzionali di un servizio. La descrizione delle funzionalità permette di definire quali sono le trasformazioni subite dall'informazione in ingresso e il cambiamento di stato del sistema che ne consegue. Il metodo utilizzato per descrivere ogni singola funzionalità esportata dal servizio è quello di definire gli *input*, gli *output*, le *precondizioni* e gli *effetti* (IOPE) della stessa. Le proprietà non funzionali sono espresse mediante opportune categorizzazioni definite in ontologie specifiche. Ad esempio alcune proprietà non funzionali di riferimento relative a persone e aziende (nome, telefono, email, ...) sono definite dall'ontologia *Actor*⁸. Oltre alle proprietà non funzionali di riferimento OWLS include una lista espandibile di proprietà espresse come *service parameters*. OWLS non definisce una cardi-

⁸<http://www.daml.org/service/owl-s/1.1/ActorDefaults.owl>

nalità nella relazione che lega un servizio ed il *Service Profile*, quindi un servizio può avere zero o più profili associati. In Figura 3.29 è proposto un stralcio della definizione del *Service Profile* del servizio di riferimento per lo scenario selezionato.

```

...
<process:Input rdf:ID="nome">
  <process:parameterType
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://www.lego-lab.org/inoassa/owl-s/apss#Nome
  </process:parameterType>
  <process:parameterType
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://www.w3.org/2002/07/owl#Thing
  </process:parameterType>
  <rdfs:label
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">nome</rdfs:label>
</process:Input>
<process:Input rdf:ID="oognome">
  ...
</process:Input>
<process:Input rdf:ID="dataNasoita">
  ...
</process:Input>
<process:Output rdf:ID="anagrafiaAssistito">
  <process:parameterType
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://www.lego-lab.org/inoassa/owl-s/apss#Assistito
  </process:parameterType>
  <process:parameterType
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://www.w3.org/2002/07/owl#Thing
  </process:parameterType>
  <rdfs:label
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    anagrafiaAssistito
  </rdfs:label>
</process:Output>
<process:Output rdf:ID="listaEsenzione">
  ...
</process:Output>
<process:Output rdf:ID="medicoDiBase">
  ...
</process:Output>
<profile:Profile
  rdf:ID="IdentifioAssistitoProcess">
  ...
  <profile:serviceName>
    AnagrafeSanitariaService
  </profile:serviceName>
  <profile:textDescription>
    Servizio di anagrafe sanitaria
  </profile:textDescription>
  <profile:contactInformation>
    <profile:Actor rdf:ID="AnagrafeSanitaria">
      </profile:Actor>
    </profile:contactInformation>
    <profile:hasInput rdf:resource="#oognome"/>
    <profile:hasInput rdf:resource="#nome"/>
    <profile:hasInput rdf:resource="#dataNasoita"/>
    <profile:hasOutput rdf:resource="#listaEsenzione"/>
    <profile:hasOutput rdf:resource="#anagrafiaAssistito"/>
    <profile:hasOutput rdf:resource="#medicoDiBase"/>
  ...
</profile:Profile>
...

```

Figura 3.29: OWLS del profilo di servizio per lo scenario di riferimento

Il *Service Model* descrive come funziona il servizio che in questo caso viene descritto come un processo. In sostanza il *Service Model* coincide con il *Process Model*. Il *Process Model* descrive le funzionalità di un servizio assieme ai processi di cui esso si compone. La descrizione delle funzionalità, come per il *Service Profile*, definisce quali sono le trasformazioni subite dall'informazione in ingresso e il cambiamento di stato del sistema espressi in termini di IOPE. I processi

componenti il servizio vengono distinti in *Atomic*, *Simple* e *Composite*. Sono definiti *Atomic* tutti quei processi che non hanno sottoprocessi e sono eseguiti in un singolo step dal punto di vista del fruitore. Sono definiti *Simple* tutti quei processi che non sono visibili al fruitori e che vengono eseguiti in un singolo step dal punto di vista del provider. Infine sono definiti *Composite* tutti quei processi che sono scomponibili in chiamate ad altri processi. Dei processi *Atomic* e *Simple* è necessario dare solo una descrizione IOPE, mentre per i *Composite* è necessario specificare anche il *control flow*. OWLS definisce che un servizio può avere al più un *Service Model*. In Figura 3.30 è proposto un stralcio della definizione del *Service Model* del servizio di riferimento per lo scenario selezionato.

```

...
<process:Input rdf:ID="nome">
  <process:parameterType
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://www.lego-lab.org/inoassa/owl-s/apss#Nome
  </process:parameterType>
  <process:parameterType
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://www.w3.org/2002/07/owl#Thing
  </process:parameterType>
  <rdfs:label
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">nome</rdfs:label>
</process:Input>
<process:Input rdf:ID="oognome">
  ...
</process:Input>
<process:Input rdf:ID="dataNasoita">
  ...
</process:Input>
<process:Output rdf:ID="anagrafiaAssistito">
  <process:parameterType
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://www.lego-lab.org/inoassa/owl-s/apss#Assistito
  </process:parameterType>
  <process:parameterType
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://www.w3.org/2002/07/owl#Thing
  </process:parameterType>
  <rdfs:label
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    anagrafiaAssistito
  </rdfs:label>
</process:Output>
<process:Output rdf:ID="listaEsenzione">...</process:Output>
<process:Output rdf:ID="medicoDiBase">...</process:Output>
<process:AtomicProcess
  rdf:ID="IdentifioAssistitoProcess">
  ...
  <process:hasInput rdf:resource="#oognome"/>
  <process:hasInput rdf:resource="#nome"/>
  <process:hasInput rdf:resource="#dataNasoita"/>
  <process:hasOutput rdf:resource="#listaEsenzione"/>
  <process:hasOutput rdf:resource="#anagrafiaAssistito"/>
  <process:hasOutput rdf:resource="#medicoDiBase"/>
  ...
</process:AtomicProcess>
...

```

Figura 3.30: OWLS del modello di servizio per lo scenario di riferimento

Il *Service Grounding* descrive come un servizio deve essere chiamato, permette in questo senso di definire il legame tra la specifica astratta (definita in OWLS) e quella concreta del servizio (definita ad esempio in WSDL). In definitiva il *Service Grounding* associa ad ogni *Atomic Process* un'istanza di servizio concreta. Un Web Service può avere più *Service Grounding* ma un *Service Grounding* si riferisce ad una singola istanza di servizio. In Figura 3.31 è propo-

sto un stralcio della definizione del *Service Grounding* del servizio di riferimento per lo scenario selezionato:

```

<...
<grounding:WsdAtomicProcessGrounding
  rdf:ID="IdentifioaAssistitoAtomicoProcessGrounding">
  <grounding:wsdOutput>
    <grounding:WsdOutputMessageMap>
      <grounding:wsdOutputMessagePart
        rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
        http://www.lego-lab.../AnagrafeSanitaria.wsdl#medicoDiBase
      </grounding:wsdOutputMessagePart>
      <grounding:owlsParameter rdf:resource="#medicoDiBase" />
    </grounding:WsdOutputMessageMap>
  </grounding:wsdOutput>
  ...
  <grounding:wsdVersion
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    1.0
  </grounding:wsdVersion>
  <grounding:wsdInput>
    <grounding:WsdInputMessageMap>
      <grounding:wsdInputMessagePart
        rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
        http://www.lego-lab.../AnagrafeSanitaria.wsdl#dataNasoita
      </grounding:wsdInputMessagePart>
      <grounding:owlsParameter rdf:resource="#dataNasoita" />
    </grounding:WsdInputMessageMap>
  </grounding:wsdInput>
  ...
  <grounding:wsdOutputMessage
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://it.tn.apss/VisuraDatiSanitari/#risposta_RiohiestaRispostaSinorona_IdentifioaAssistito_Msg
  </grounding:wsdOutputMessage>
  <grounding:wsdInputMessage
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://it.tn.apss/VisuraDatiSanitari/#riohiesta_RiohiestaRispostaSinorona_IdentifioaAssistito_Msg
  </grounding:wsdInputMessage>
  <grounding:wsdDocument
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://www.lego-lab.../AnagrafeSanitaria.wsdl
  </grounding:wsdDocument>
  <grounding:wsdOperation>
    <grounding:WsdOperationRef>
      <grounding:portType
        rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
        http://www.lego-lab.../AnagrafeSanitaria.wsdl#VisuraDatiSanitariPort
      </grounding:portType>
      <grounding:operation
        rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
        http://www.lego-lab.../AnagrafeSanitaria.wsdl#IdentifioaAssistito
      </grounding:operation>
    </grounding:WsdOperationRef>
  </grounding:wsdOperation>
</grounding:WsdAtomicProcessGrounding>

```

Figura 3.31: OWLS del grounding di servizio per lo scenario di riferimento

WSMO/WSML

WSMO o Web Service Modeling Ontology è un'ontologia sviluppata per supportare il dispiegamento e l'interoperabilità di Web Service semantici.

Come riportato in Figura 3.32 WSMO è basato su quattro elementi concettuali *Ontologie*, *Goal*, *Web Service* e *Mediatori*. L'insieme dei concetti permette di descrivere la *semantica di riferimento* gli *obiettivi* e gli aspetti relativi al *com-*

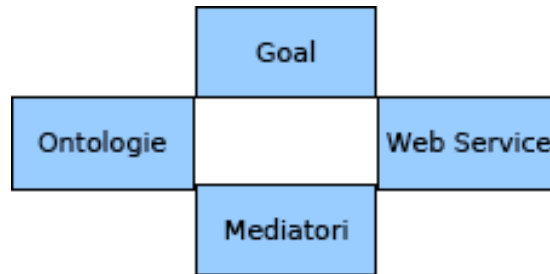


Figura 3.32: Elementi concettuali di WSMO

portamento del servizio; inoltre aggiunge il concetto di mediatore che permette di *affrontare eventuali problemi di interoperabilità* tra *peer* eterogenei.

Le *Ontologie* vengono definite al metalivello, in questo senso WSMO rimane agnostico rispetto alle scelte sul linguaggio di specifica delle stesse. Oltre a definire gli elementi di una ontologia al metalivello, per ogni ontologia dichiarata WSML definisce un *Mediatore* utilizzato per importare l'ontologia stessa.

Di un *Web Service* sono specificate le *Capability* – ossia le funzionalità – la *interface* – ossia il protocollo di interazione e funzionamento – ed inoltre è possibile specificare ulteriori proprietà non funzionali che caratterizzano alcuni aspetti specifici del servizio. Ogni *Capability* descrive in modo formale gli stati precedenti e successivi all'esecuzione della funzionalità; in particolare di una specifica funzione sono individuate le *precondizioni*, le *postcondizioni* e le *assunzioni* valide a monte e a valle dell'esecuzione della stessa. La *Interface* esprime il protocollo di interazione in termini di *Coreografia* ed *Orchestrazione*. La *Coreografia* descrive le informazioni necessarie al *service requester* per comunicare con il servizio, mentre l'*Orchestrazione* descrive come il servizio funziona dal punto di vista del *service provider*. La vista di orchestrazione e coreografica sono espresse in un linguaggio basato su *Abstract State Machines* che abbina ad un'alta flessibilità nel modellare i sistemi un ben conosciuto modello formale.

I *Goal* sono espressi in termini di obiettivi che il *service requester* può avere nel momento in cui consulta un servizio e possono essere espressi tramite l'utilizzo di tutti gli elementi definiti per descrivere un *Web Service*. Ad esempio è possibile definire un *Goal* in termini di *precondizioni*, *postcondizioni* ed *effetti*.

WSML o *Web Service Modeling Language* è un linguaggio basato sul modello concettuale di WSMO. L'obiettivo di WSML è quello di proporre un framework coerente che punta a definire gli aspetti e i concetti espressi in WSMO con linguaggi formali.

In Figura 3.33 viene riportato uno stralcio dell'ontologia dello scenario di riferimento modellata in WSML che riporta la definizione del concetto di *Persona* delle *Capability* del servizio.

```

...
concept Persona
  nonFunctionalProperties
    do:description hasValue "concetto di persona"
  endNonFunctionalProperties
  hasName ofType foaf#Nome
  hasCognome ofType foaf#Cognome
  hasDataNasoita ofType foaf#DataDiNasoita
  ...
concept Assistito subConceptOf {Persona}
...
capability
  sharedVariables ?assistito
  precondition
    nonFunctionalProperties
      do:description hasValue "Richiesta anagrafica assistito"
    endNonFunctionalProperties
  definedBy
    ?assistito memberOf Assistito
      and ?assistito[hasNome hasValue ?nome]
      and ?assistito[hasCognome hasValue ?cognome]
      and ?assistito[hasDataNasoita hasValue ?dataDiNasoita] .
...

```

Figura 3.33: WSMML del servizio per lo scenario di riferimento, limitatamente a Persona e Capability

WSDLS

WSDLS è un'estensione proposta al linguaggio WSDL intesa a permettere l'integrazione di annotazione di tipo semantico ai costrutti standard definiti dal linguaggio stesso. Rispetto ad altre tecniche, WSDLS rimane agnostico rispetto alla scelta del linguaggio con cui definire l'ontologia del dominio di riferimento. In questo contesto WSDLS si propone l'obiettivo di mettere a disposizione dell'utente un insieme di strumenti per "legare" gli elementi standard definiti dal linguaggio di specifica ai concetti ed alle classi definiti nell'ontologia del dominio di riferimento. In particolare WSDLS permette di definire gli *input*, gli *output*, le *precondizioni* e gli *effetti* legati alle varie funzionalità di un servizio web. In Figura 3.34 viene presentato dal punto di vista logico come i diversi elementi di WSDL sono annotati e legati ad un'ontologia del dominio di riferimento.

In Figura 3.35 viene presentata l'annotazione del WSDL dello scenario di riferimento.

L'approccio che caratterizza WSDLS non si propone di definire un unico linguaggio che cattura in unica vista organica i diversi aspetti necessari alla modellazione di un servizio. L'approccio proposto si propone di utilizzare linguaggi standard – o eventualmente estensioni a linguaggi standard – già di uso comune nella comunità di sviluppatori per modellare aspetti specifici del servizio. In questo contesto ad esempio la parte di processo potrebbe essere modellata con BPEL4WS o con una sua estensione che supporti l'annotazione semantica delle varie attività del processo.

SAWSDL

SAWSDL o Sematic Annotation for WSDL and Schema definisce come arricchire di annotazioni semantiche varie parti di un documento WSDL seguendo le

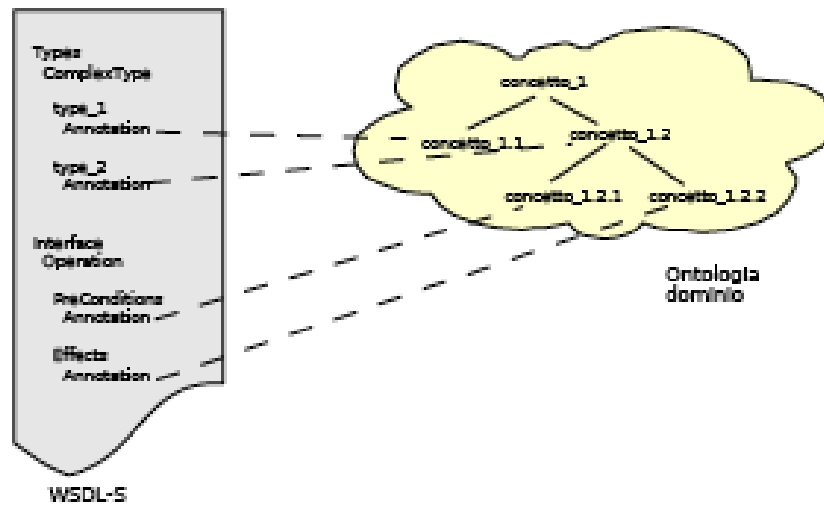


Figura 3.34: WSDLS

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://it.tn.apss/VisuraDatiSanitari/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name="interfaociaImplementativaErogatore"
  xmlns:rappOntology="http://www.lego-lab.org/incassa/owl-s/apss"
  targetNamespace="http://it.tn.apss/VisuraDatiSanitari/">

  <wsdl:types>
    <xsd:schema targetNamespace="http://it.tn.apss/VisuraDatiSanitari/"
      xmlns:tns="http://it.tn.apss/VisuraDatiSanitari/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
      ...
      <xsd:complexType name="AnagrafiaAssistito"
        wssem:modelreference="apssOntology#Assistito">
        <xsd:sequence>
          <xsd:element name="oodSanitario"
            type="tns:CodiceSanitarioNazionale"/>
          <xsd:element name="oodFisuale" type="tns:CodiceFisuale"/>
          ...
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="AnagrafiaMedicoDiBase"
        wssem:modelreference="apssOntology#MedicoDiBase">
        <xsd:sequence>
          <xsd:element name="oodFisuale" type="tns:CodiceFisuale"/>
          <xsd:element name="oognome" type="tns:Cognome"/>
          ...
        </xsd:sequence>
      </xsd:complexType>
      ...
    </xsd:schema>
  </wsdl:types>
  ...
  <wsdl:interface name="VisuraDatiSanitari_PT">
    <wsdl:operation name="IdentifioaAssistito" pattern="wsdl:in-out">
      <wsdl:input
        messageLabel="RichiestaIdentifioaAssistito"
        element="tns:richiesta_RichiestaRispostaSinorona_IdentifioaAssistito_Msg" />
      <wsdl:output
        messageLabel="RispostaIdentifioaAssistito"
        element="tns:risposta_RichiestaRispostaSinorona_IdentifioaAssistito_Msg" />
      <wssem:precondition name="PersonaIsResidentePrecond"
        wssem:modelreference="apssOntology#PeronaIsResidente"/>
    </wsdl:operation>
  </wsdl:interface>
</wsdl:definitions>

```

Figura 3.35: WSDLS del servizio per lo scenario di riferimento

raccomandazioni definite nel WSDL 2.0/WSDL 1.1 extensibility framework. Le annotazioni semantiche sono definite attraverso l'utilizzo di attributi aggiuntivi rispetto alla grammatica standard di WSDL che permettono di legare gli elementi propri del linguaggio all'ontologia di riferimento, in questo molti dei concetti introdotti sono mutuati da WSDLS.

SAWSDL si concentra sull'annotazione della parte astratta definita dal WSDL e non entra nella specifica dell'implementazione dello stesso, in particolare sono definiti tre attributi: *modelReference*, *liftingSchemaMapping* e *loweringSchemaMapping*. L'attributo *modelReference* permette definire l'associazione tra un componente WSDL ed un concetto definito nell'ontologia di riferimento. Gli attributi *liftingSchemaMapping* e *loweringSchemaMapping* sono aggiunti alla parte di definizione del XML Schema e consentono di specificare la relazione tra il documento XML e l'ontologia di riferimento.

In Figura 3.36 viene presentata l'annotazione SAWSDL del WSDL dello scenario di riferimento.

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://it.tn.apss/VisuraDatiSanitari/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name="interfaooiaImplementativaErogatore"
  xmlns:appsOntology="http://www.lego-lab.org/incassa/owl-s/apss"
  targetNamespace="http://it.tn.apss/VisuraDatiSanitari/">

  <wsdl:types>
    <xsd:schema targetNamespace="http://it.tn.apss/VisuraDatiSanitari/"
      xmlns:tns="http://it.tn.apss/VisuraDatiSanitari/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
      ...
      <xsd:complexType name="AnagrafiaAssistito"
        sawSDL:modelReference="appsOntology#Assistito">
        <xsd:sequence>
          <xsd:element name="oodSanitario"
            type="tns:CodiceSanitarioNazionale"
            sawSDL:modelReference="appsOntology#CodiceSanitarioNazionale"/>
          <xsd:element name="oodFiscale" type="tns:CodiceFiscale"/>
          ...
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="AnagrafiaMedicoDiBase"
        sawSDL:modelReference="appsOntology#MedicoDiBase">
      </xsd:complexType>
    </xsd:schema>
  </wsdl:types>
  ...
  <wsdl:interface name="VisuraDatiSanitari_PT"
    sawSDL:modelReference="appsOntology#RicercaAssistito">
    <wsdl:operation name="IdentificiaAssistito" pattern="wsdl:in-out">
      <wsdl:input
        messageLabel="RichiestaIdentificiaAssistito"
        element="tns:richiesta_RichiestaRispostaSinorona_IdentificiaAssistito_Msg" />
      <wsdl:output
        messageLabel="RispostaIdentificiaAssistito"
        element="tns:risposta_RichiestaRispostaSinorona_IdentificiaAssistito_Msg" />
    </wsdl:operation>
  </wsdl:interface>
</wsdl:definitions>

```

Figura 3.36: SAWSDL del servizio per lo scenario di riferimento gi rappresentato in WSDL

Come per il precedente l'approccio che caratterizza SAWSDL non si propone di definire un unico linguaggio che catturi in unica vista organica i diversi aspetti necessari alla modellazione di un servizio. L'approccio proposto si propone di

utilizzare linguaggi standard – o eventualmente estensioni a linguaggi standard – già di uso comune nella comunità di sviluppatori per modellare aspetti specifici del servizio.

3.6.4 Conclusioni

In questo documento è stata presentata una vista d'insieme dei principali linguaggi e delle tecniche per modellare informazioni di carattere semantico nell'ambito di un sistema informativo basato sui servizi web.

Nei linguaggi presentati sono distinguibili due tipi di approccio alla modellazione semantica dei servizi web.

Nel caso di OWLS e WSMO sono modellati a livello ontologico sia il contesto del dominio di riferimento che gli aspetti legati ai servizi. Nel processo di modellazione semantica supportato in questo contesto alcune delle informazioni sono contenute sia nel documento WSDL che nella descrizione semantica fatta nei linguaggi di riferimento.

Nel caso di WSDLS e SAWSDL si punta a definire un'ontologia del solo dominio di riferimento e – tramite l'estensione del documento WSDL – a legare con delle annotazioni i vari elementi dell'interfaccia alla semantica del dominio.

3.7 Ontologie, servizi e metadati

Parallelamente allo sviluppo di servizi web semantici, il campo tradizionale dei metadati ha sviluppato sistemi e repertori propri, che si pongono adesso sia in alternativa, sia come materiale riusabile per lo sviluppo di ontologie e servizi semantici. Nella prossima sezione, una overview dell'uso dei metadati nel dominio dell'interoperabilità è associata a un'analisi delle motivazioni per cui è necessario passare decisamente a una gestione semantica dei servizi.

3.8 Metadati

Basato su materiale contribuito da:
Università di Camerino e-Government Research Group

3.8.1 Introduzione

Il termine “metadato” secondo il gergo tecnico comune rappresenta l'insieme delle informazioni relative ai dati, sui dati, intorno ai dati. Il lemma derivante dall'inglese: “metadata” (pl.), è a sua volta formato dal termine greco “meta” (attinente) e dal latino “data”, pl. di “datum” (informazione). più propriamente con il termine metadati si fa riferimento a delle proprietà di un insieme di dati.

In letteratura, inoltre, sono presenti una serie di definizioni di seguito riportate.

- “Machine-understandable information about Web resources or other things” (Tim Berners-Lee, 1997).
- “Data associated with objects which relieves their potential users of having to have full advance knowledge of their existence or characteristics. A user might be a program or a person” (Lorcan Dempsey & Rachel Heery, 1998).
- “Structured data about resources that can be used to help support a wide range of operations” (Michael Day, 2001).
- “Structured data about data” (Dublin Core Working Group, 2003).

Resta chiaro che la distinzione tra dati e metadati non è intrinseca, ma dipende dal contesto di utilizzo. Gli stessi metadati sono dati, e perciò possono essere memorizzati come dati, (nella risorsa stessa o in un'altra risorsa), e possono essere descritti da altri metadati, e così via. Il numero di metalivelli da specificare dipende quindi dalle caratteristiche delle applicazioni e dalle specifiche esigenze.

Un esempio tipico di metadati è costituito dalla scheda del catalogo di una biblioteca, la quale contiene informazioni circa il contenuto e la posizione di un libro: questi sono dati riguardanti i dati nel libro cui si fa riferimento mediante la scheda. Altro contenuto tipico dei metadati può essere rappresentato dalla fonte, dall'autore dell'insieme di dati descritto o dalle modalità d'accesso con le eventuali limitazioni.

Da sottolineare è il ruolo che i metadati hanno assunto con l'avvento del World Wide Web ed in particolare in relazione alla necessità sempre più incombente di trovare informazioni utili nella massa di informazione disponibili. In questo contesto, la capacità dei metadati di essere compresi oltre che dagli uomini anche dalle macchine, risulta essere un'arma vincente per la riduzione di tempi e dei costi legati alla ricerca/comprendimento delle informazioni.

Nelle PA, in particolare, le informazioni sono l'output primario dei processi di e-Government. Infatti, il volume delle informazioni prodotte dalle Pubbliche Amministrazioni sta crescendo con un tasso di sviluppo piuttosto elevato, in questo contesto è essenziale portare ordine in questa massa di informazioni al fine di garantire la loro disponibilità in modo continuativo. E' essenziale, che sia l'accesso alle informazioni, che la capacità dei cittadini a localizzare quelle che soddisfano i propri interessi deve essere supportata quanto possibile. A tal fine si evince la proposta di introdurre degli standard di metadati in grado di aiutare gli utenti nel ritrovamento delle informazioni relative ai servizi pubblici in modo facile e veloce. I metadati rappresentano quindi un fattore chiave per lo sviluppo e la fornitura di informazioni e servizi on-line nel settore pubblico.

Questo report che introduce al mondo dei metadati è così strutturato: nella seconda sezione si discutono le caratteristiche ed i possibili utilizzi dei metadati, nella terza sezione si introducono i modelli di metadati presenti in letteratura, nella quarta i tool utili alla gestione dei metadati, mentre nella quinta si introduce all'importanza dei metadati nella Pubblica Amministrazione. Nella sesta sezione si presenta l'architettura dei dati RDF ed infine la settima sezione conclude il lavoro.

3.8.2 Modelli di Metadati

Di seguito mostriamo alcuni standard di metadati presenti in letteratura si evidenzia chiaramente che comunità diverse impiegano diversi vocabolari di metadati. E' opinione comune che la prima impressione che si ha, quando si comincia ad interessarsi al tema della modellazione dell'informazione, è che gli standard presenti siano perfino troppi. Di fatto oltre a quelli di seguito elencati, se ne potrebbero citare molti altri.

I Precursori

Platform for Internet Content Selection (PICS) Per ragioni storiche deve essere citato lo standard PICS⁹ (Platform for Internet Content Selection, che risale al 1996 ed ha come scopo principale la condivisione e l'interscambio di valutazioni su siti e pagine Web. PICS si basa sull'uso di etichette trasportate negli header HTTP o inserite all'interno di pagine HTML come marcatori "META".

⁹<http://www.w3.org/PICS/>

MAchine Readable Cataloguing record (MARC) Lo standard MARC¹⁰ è stato sviluppato alla fine degli anni 60 alla Library of Congress (USA) per promuovere la condivisione di cataloghi tra biblioteche. È diventato ad oggi un formato quasi standard. Esso utilizza un sistema di numeri, lettere e simboli all'interno del record per individuare i diversi tipi di informazione.

Metadati di applicazione generale

Dublin Core (DC) Lo standard Dublin Core¹¹ della Dublin Core Metadata Initiative, ad oggi più diffuso, è caratterizzato da una estrema semplicità in grado tuttavia di essere esteso per applicazioni più complesse. Esso consiste in un massimo di 15 elementi descrittivi (metatag) i quali possono essere categorizzati come di seguito descritto:

- elementi relative al contenuto della risorsa (title, subject, description, source, language, relation, e coverage);
- elementi relativi alla proprietà intellettuale (creator, publisher, contributor, e rights);
- elementi relativi principalmente alle caratteristiche intrinseche al documento, istanziazione della risorsa (date, type, format, e identifier).

Lo standard Dublin Core ha ricevuto l'approvazione NISO (National Information Standard Organisation) e dall'ottobre del 2001 è ufficialmente una standard ANSI (American National Standards Institute) (Z39.85-2001).

Elemento	Etichetta	Descrizione
Titolo	Title	Il nome dato alla risorsa, di norma dal Creatore o Editore
Autore o Creatore	Creator	La persona o organizzazione che ha la responsabilità principale della produzione del contenuto intellettuale della risorsa. Per esempio, autori nel caso di documenti scritti, artisti, fotografi, o illustratori nel caso di risorse visive
Soggetto e Parole chiave	Subject	L'argomento della risorsa; tipicamente, il soggetto sarà espresso da parole chiave o frasi che descrivono l'argomento o il contenuto della risorsa

¹⁰<http://www.loc.gov/marc/umb/>

¹¹<http://www.dublincore.org>

Global Information Locator Service (GILS) L'obiettivo principale di GILS¹² (Global Information Local Service) è quello di facilitare il ritrovamento delle informazioni di ogni tipo, con tutti i mezzi, in tutte le lingue e soprattutto nel tempo. La strategia seguita è quella di supportare l'evolvemento di uno standard internazionale per la ricerca delle informazioni. GILS definisce uno standard aperto, a bassi costi, e scalabile così che il governo, le imprese e altre organizzazioni possano aiutare o essere aiutati nella ricerca di informazioni.

Metadati per specifici domini disciplinari

Multimedia Content Description Interface (MPEG7) Per quel che riguarda i contenuti audiovisivi, la descrizione delle risorse multimediali può essere affidata allo standard MPEG-7, che è nato nel settembre 2001 e si occupa di definire come tali descrizioni debbano essere formulate e soprattutto interpretate. Ad esempio, in una stessa scena (intesa come una data sequenza di fotogrammi), è possibile etichettare informazioni caratterizzate da un alto livello di dettaglio grafico e da un basso contenuto informativo (es: un quadrato giallo di dimensioni 50 x 50 pixel) oppure informazioni molto più sintetiche, ma a maggior contenuto semantico (es: il ritorno dell'eroe). Il livello di astrazione è influenzato dal modo in cui tali informazioni sono ottenute dal sistema. Il formato di codifica dei vari media, i diritti di utilizzo dei contenuti, la data di produzione dell'oggetto sono altri esempi di metadati caratteristici di MPEG-7 che possono essere aggiunti per descrivere meglio il contenuto audiovisivo.

Text Encoding Initiative (TEI) TEI¹³ (Text Encoding Initiative) rappresenta una codifica standard per i documenti testuali usata per descrivere la struttura fisica e logica di materiale testuale allo scopo di ricercare, analizzare e scambiare i dati. Un header contiene informazioni bibliografiche e l'origine precedente della codifica completa.

Encoded Archival Description (EAD) EAD¹⁴ (Encoded Archival Description) è una struttura dati standard usata per codificare ed archiviare manoscritti. Le descrizioni individuali sono variamente chiamate "aids", "guides", "handlists", o "catalogs" e seguono la sintassi dello standard Standard Generalized Markup Language (SGML).

Consortium for the Interchange of Museum Information (CIMI) Il CIMI¹⁵ (Consortium for the Interchange of Museum Information) rappresenta uno standard per lo scambio digitale di dati tra musei.

¹²<http://www.gils.net/>

¹³<http://www.tei-c.org/>

¹⁴<http://www.loc.gov/ead/>

¹⁵<http://www.cimi.org/index.html>

Visual Resources Association (VRA) Core Categories VRA¹⁶ (Visual Resources Association) viene usato per descrivere i lavori di culture visive (ad esempio pittori, scultori, costruttori) tanto bene come le loro immagini surrogate (digitali, fotomeccaniche, fotografiche) o lavori di cultura materiale e le relative immagini.

Content Standard for Digital Geospatial Metadata (CSDGM) CSDGM¹⁷ (Content Standard for Digital Geospatial Metadata) ha come obiettivo quello di fornire un insieme comune di termini e definizioni per la documentazione di dati digitali geospaziali. Lo standard stabilisce i nomi degli elementi elementari e composti al fine di essere usati per questi scopi, le definizioni di questi elementi e le informazioni circa i valori sono fornite per i dati elementari.

Online Information Exchange (ONIX) ONIX¹⁸ (Online Information Exchange) è uno standard internazionale per la rappresentazione di libri, romanzi e video prodotti in forma elettronica. Molti commercianti di libri come Amazon e Barnes & Noble usano questi metadati per trasferire informazioni circa i loro prodotti.

Friend Of a Friend: metadati per descrivere reti sociali Il Friend of a Friend¹⁹ (FOAF) riguarda la creazione di pagine Web che sono leggibili dalla macchina e descrivono le persone e le reti sociali che si realizzano tra esse. Al tempo stesso si descrivono i collegamenti tra le attività pianificate. Alcuni elementi base sono: person, name, nick, title, surname, weblog, knows, interest, ecc.

3.8.3 Caratteristiche Utilizzi e Funzioni dei Metadati

Sulla base delle funzionalità che i metadati sono in grado di gestire essi possono essere distinti principalmente in tre categorie funzionali di seguito proposte.

Descrittivi: per l'identificazione e il recupero degli oggetti digitali; sono costituiti da descrizioni normalizzate dei documenti digitali nativi, risiedono generalmente nelle basi dati dei sistemi di Information Retrieval all'esterno degli archivi degli oggetti digitali e sono collegati a questi ultimi tramite appositi link.

Amministrativi e gestionali: forniscono dati utili per la gestione della risorsa descritta, per le operazioni di gestione degli oggetti digitali all'interno dell'archivio.

¹⁶<http://www.vraweb.org/>

¹⁷<http://www.fgdc.gov/metadata/csdgm/>

¹⁸<http://www.bisg.org/onix/index.html>

¹⁹<http://www.foaf-project.org/>

Strutturali: descrivono la struttura interna fisica o logica dei documenti (es. introduzione, capitoli, sezioni, indice di un libro) e le loro relazioni fra le varie parti degli oggetti digitali.

I metadati possono poi essere distinti anche sulla base della loro localizzazione rispetto alla risorsa che descrivono in **embedded** o **esterni**. Nel primo caso essi sono inclusi nell'oggetto, mentre nel secondo caso essi sono esterni all'oggetto ed archiviati a parte come link all'oggetto descritto.

Le principali funzioni che essi si prefiggono di realizzare sono quelle di:

- gestire le risorse fornendo la loro identificazione a favore della loro presentazione e recupero;
- consentire l'organizzazione, la localizzazione, la gestione e le statistiche;
- favorire l'interoperabilità e l'integrazione di risorse simili;
- facilitare l'archiviazione e la conservazione delle risorse.

Inoltre, si evidenziano i seguenti principi generali che nascono dall'attività di costruzione dei metadati: la modularità, l'estensione ed il raffinamento. La **modularità** o granularità dei metadati è la chiave per organizzare le diverse fonti, contenuti e approcci rispetto alla descrizione della risorsa. Questo permette a chi deve realizzare uno schema di metadati di creare nuove aggregazioni fra più schemi e di stabilirne di nuovi, tenendo conto di quelli già realizzati, che sono giudicati best practice, piuttosto che reinvestirne sulla creazione di altri. L'**estensione** nella costruzione dei metadati permette di partire dalla nozione di uno schema già definito ed aggiungere ulteriori elementi per particolari applicazioni o domini. Dall'altra parte queste estensioni non devono compromettere l'interoperabilità dello schema. Nei vari profili di applicazione è auspicabile, inoltre, che il grado di dettaglio degli elementi vari in base alle diverse esigenze secondo processi diversi di **raffinazione**. Per ogni particolare applicazione deve essere scelto uno schema appropriato.

3.8.4 Tool per la gestione dei metadati

Di seguito un elenco di tool che possono essere presi in considerazione per la creazione dei metadati Dublin Core. Essi tuttavia rappresentano solo una parte degli strumenti attualmente disponibili.

- <http://www.ukoln.ac.uk/metadata/dcdot/>
- <http://www.lub.lu.se/cgi-bin/nmdc.pl>
- <http://metadata.net/dstc>
- <http://www.mkdoc.org/>
- <http://www.library.kr.ua/dc/dcreditunie.html>

- http://www.webpossible.com/utilidades/photo_rdf_generator_en.html
- <http://www.describethis.com/>
- <http://purl.oclc.org/scorpion>
- <http://www.library.kr.ua/dc/lookatdce.html>

3.8.5 Metadati soluzioni a supporto della Cooperazione Applicativa SPCoop

In generale si presentano modelli di interoperabilità semantica nella cooperazione applicativa. Dal punto di vista della semantica, gli approcci alla cooperazione applicativa si caratterizzano per:

1. la condivisione dei vocabolari (condivisi o separati)
2. la distribuzione della logica di integrazione (centralizzata o decentralizzata)

Nella PA la cooperazione applicativa è decentralizzata in quanto le amministrazioni stringono accordi per regolare la cooperazione su base prioritaria (accordi di servizio), la condivisione dei metadati si realizza nell'ambito di accordi multilaterali, l'implementazione degli accordi di servizio è nei singoli sistemi informativi delle Amministrazioni, la logica di integrazione è nell'implementazione dei singoli sistemi informativi.

Dal punto di vista dei vocabolari si possono individuare entrambe le soluzioni: separati o condivisi. Nella prima l'erogatore del servizio pubblica il proprio modello concettuale, i fruitori traducono poi il modello di servizio nei termini della propria applicazione, la semantica del sistema di cooperazione è data dal fruitore sotto forma di regole di traduzione del modello dell'erogatore. Nei modelli decentralizzati con vocabolario condiviso, invece, erogatori e fruitori dei servizi condividono un vocabolario di metadati, ciascuno interpreta il vocabolario condiviso rispetto alla propria concettualizzazione, la semantica del sistema di cooperazione è dato da ognuno sotto forma di regole di interpretazione del vocabolario condiviso.

Si considera quindi che avere un insieme di termini condiviso permette di evitare la costruzione di una matrice di traduzione tra modelli concettuali eterogenei, per gestire la semantica della cooperazione tra N sistemi, la cui dimensione può collocarsi nell'ordine di N^2 . Al tempo stesso non ci si affida alla correttezza delle ipotesi semantiche del traduttore per gestire l'accuratezza delle traduzioni. La gestione della cooperazione applicativa con metadati standard prevede quindi di gestire la semantica della cooperazione tra N sistemi attraverso un insieme di interpretazioni la cui dimensione è N . L'accuratezza di una interpretazione è legata alla comprensione della semantica condivisa da parte dell'interprete per la quale è possibile ipotizzare una più agevole verifica.

Quindi a fronte di un adeguato investimento, la standardizzazione dei metadati può portare alla cooperazione applicativa notevoli benefici tra cui una

maggior accuratezza nella gestione degli aspetti semantici della cooperazione applicativa, flessibilità, semplicità e robustezza. Si possono poi trovare una serie di vantaggi quantitativi relativamente alla reattività al cambiamento (in termini di riduzione dei tempi per la ricezione di leggi e normative), alla riduzione dei costi di integrazione ed all'incremento della riusabilità delle soluzioni in quanto le applicazioni potrebbero essere basate su modelli concettuali omogenei.

PA diverse possono avere linguaggi con una definita semantica ma con sintassi diverse. A tutt'oggi i sistemi informativi evidenziano la mancanza di una sintassi condivisa fattore di inibizione per lo sviluppo della cooperazione applicativa. La standardizzazione dei metadati risulta quindi essere il modo migliore per definire la sintassi in sistemi di cooperazione decentralizzati come quelli della PA.

3.8.6 Metadati e soluzioni di e-Government

Nel dominio di e-Government una serie di attività sono state attivate al fine di introdurre soluzioni ricche a livello semantico per quanto riguarda i metadati all'interno della PA. Prima su tutte l'Australia con lo standard AGIS (Australian Government Locator Service) basato sul DC con 4 elementi addizionali, che hanno adottato per acquisire informazioni sui servizi e sui documenti della Pubblica Amministrazione. Analoghi schemi, basati o largamente compatibili col DC sono stati sviluppati in Canada, in Irlanda e in alcuni stati degli USA oltre che in UK e Danimarca (cfr. allegato). Nascono quindi una serie di standard di metadati (diversi per ogni Paese) che puntano al raggiungimento degli obiettivi sopra proposti.

In particolare, si evidenzia la nascita, tra gli altri, di un gruppo e-Government (nel contesto delle Dublin Core Metadata Initiative) che si occupa della applicazione del DC e mira pertanto ad identificare elementi comuni nelle applicazioni di e-Government in corso ed a promuovere l'uso del DC in questa comunità. Il gruppo di lavoro del DCMI propone per il profilo applicativo dell'informazione pubblica un nuovo elemento, il dc.audience, ripreso dal profilo elaborato dal gruppo di lavoro per il settore education (altro gruppo DCMI), e una serie di nuovi qualificatori dell'elemento dc.date, dc.relation ("isBasedOn" e "isBasisFor") e soprattutto del dc.rights, con l'introduzione dei qualificatori "securityClassification", "previousSecurityClassification", "accessRight", "copyright". Infine, il qualificatore del dc.type "aggregationLevel". Ancora a livello di bozza è, invece, l'identificazione delle caratteristiche per la descrizione dei servizi forniti dalle Pubbliche Amministrazioni.

3.8.7 Architettura dei Dati: RDF

Quello che ci si chiede a questo punto della trattazione è individuare delle metodologie di modellazione di concetti e semantiche, nonché i relativi formalismi di rappresentazione, più adeguati alla definizione di una base di metadati standardizzata e condivisa per la PA, sulla base del miglior rapporto tra espressività e semplicità d'uso.

Si può pensare qui di prendere in considerazione l’RDF come uno dei linguaggi attualmente disponibili al fine di far comprendere alle macchine il significato delle informazioni scambiate sul Web fornendo un modello molto semplice e flessibile capace di codificare tabelle, alberi, ecc. RDF come HTML, permette di creare dei link tra risorse pubblicate sul Web ed al contrario dei comuni ipertesti permette di etichettarli, al fine di indicare esplicitamente le relazioni che intercorrono tra gli oggetti. Inoltre, questi link fanno riferimento ad oggetti tipizzati (metadati) e permettono così di dare significato ad una risorsa.

Immaginiamo di voler tener traccia del fatto che qualcuno di nome Mario Rossi ha creato una certa pagina Web (<http://www.example.org/index.html>); se dovessimo descrivere questo fatto in linguaggio naturale potremmo utilizzare una semplice frase: “<http://www.example.org/index.html> ha un autore il cui valore è Mario Rossi”. Alcune parti della frase sono state sottolineate per identificare:

- ciò che si vuole descrivere (in questo caso la pagina) [soggetto];
- la specifica proprietà (autore) di ciò che si vuole descrivere [predicato];
- ciò che vogliamo utilizzare per assegnare un valore alla proprietà (chi è l’autore) di ciò che si vuole descrivere [oggetto].

Per rendere frasi simili a quella precedente processabili dalla macchina occorrono tuttavia (i) degli identificativi “machine-processable” che permettano di identificare il soggetto, l’oggetto ed il predicato in una frase senza nessuna possibilità di confusione ed (ii) un formato “machineprocessable” per rappresentare e scambiare le frasi tra macchine. Il Web di oggi già dispone di entrambi i meccanismi: (i) gli Uniform Resource Identifier (URI) disponibili per identificare qualunque cosa di cui si voglia parlare in uno statement RDF e (ii) l’Extensible Markup Language (XML) come formato ideale per rappresentare e scambiare degli statement RDF.

Per scrivere quindi la frase sopra indicata occorre quindi assegnare un URI a tutte (o quasi) le parti sottolineate della frase. Nello statement RDF di esempio:

- il soggetto è già un URI;
- il valore (l’oggetto) “Mario Rossi” potrebbe essere identificato dall’URI `URN:org:example:staffid:85740`;
- per il predicato “autore” si può utilizzare l’URI <http://purl.org/dc/elements/1.1/creator>.

Si ottiene quindi il seguente risultato:

“<http://www.example.org/index.html>
ha un <http://purl.org/dc/elements/1.1/creator>
il cui valore è `URN:org:example:staffid:85740`”.

E' interessante a questo punto osservare l'utilizzo del tag Creator appartenente agli elementi Dublin Core per identificare la proprietà autore. Al tempo stesso si potrebbe pensare di estendere lo stantement indicato descrivendo ulteriormente la risorsa magari indicando la sua età, mail, ecc. Anche in questo caso si potrebbe far uso dei metadati definiti tra gli standard proposti precedentemente ed in particolare l'elemento e-mail dello standard Foaf. Il tutto risulterebbe come di seguito riportato “**ex:index.html** ha un **dc:creator** il cui valore è **sid:85740** ed ha un **foaf:email** il cui valore è **mailto:mrossi@example.org**”.

Si potrebbe quindi pensare ad un modello di riferimento (che tuttavia non è l'unico possibile) e prevede l'utilizzo di tre standard come layers comunicativi: (i) XML stabilisce la sintassi di base (struttura fisica), (ii) RDF definisce la struttura delle relazioni (struttura logica) e (iii) DC definisce i contenuti (struttura per i contenuti).

3.8.8 Conclusioni

Alla luce di quanto detto si evince chiaramente l'importanza della definizione di standard opportuni per la rappresentazione delle informazioni nel contesto del Sistema Pubblico di Connettività (SPC) definito come “l'insieme di strutture organizzative, infrastrutture tecnologiche e regole tecniche, per lo sviluppo, la condivisione, l'integrazione e la circolarità del patrimonio informativo della Pubblica Amministrazione, necessarie per assicurare l'interoperabilità e la cooperazione applicativa dei sistemi informatici e dei flussi informativi, garantendo la sicurezza e la riservatezza delle informazioni”. In particolare, relativamente alla cooperazione applicativa come sistema finalizzato all'interazione tra i sistemi informatici delle Pubbliche Amministrazioni per garantire l'integrazione delle informazioni e dei procedimenti amministrativi.

La standardizzazione dei metadati è il modo migliore per definire la semantica in sistemi di cooperazione decentralizzati come quelli della PA. A fronte di un adeguato investimento, la standardizzazione può portare alla cooperazione applicativa notevoli benefici sia quantitativi che qualitativi. Relativamente ai primi si evidenzia maggiore accuratezza nella gestione degli aspetti semantici della cooperazione, riduzione delle interdipendenze, trasparenza semantica dei dati ed equidistribuzione della logica applicativa. Dal lato qualitativo si evidenzia, invece, la nascita di applicazioni basate su modelli concettuali omogenei, riduzione dei tempi per la ricezione di leggi e normative ed una sostanziale riduzione dei costi di integrazione. Naturalmente si devono considerare una serie di costi come quelli di standardizzazione (produzione del metamodello, di un modello generale – ontologia – e di modelli specifici), gestione (servizi di metadati e di authoring cooperativo) ed adozione (formazione, consulenza, software specifico).

Concludendo, il linguaggio della Pubblica Amministrazione ha una semantica definita, ma a tutt'oggi i sistemi informativi della PA hanno in comune, per lo più, qualche sintassi. In questo modo la mancanza di una semantica condivisa è un fattore di inibizione per lo sviluppo della Cooperazione Applicativa nel sistema paese che deve trovare le forze per gestire il processo di standar-

dizzazione del vocabolario della PA e farne un volano per favorire l'efficienza amministrativa, i servizi al cittadini e l'economia.

Capitolo 4

Scenari applicativi per la semantica nella PA

4.1 Confronto fra scenari applicativi nella PA

NOTA: questa sezione ancora da scrivere.

4.2 Il catalogo metadati della PA piemontese

A cura di: CSI Piemonte

Il seguente contributo descrive 2 esperienze, una di tipo progettuale, l'altra sperimentativa. Le due esperienze sono titolate in grassetto come:

1. Il catalogo metadati della Pubblica Amministrazione Locale piemontese.
2. Esperimenti di estrazione e correlazione di concetti.

4.2.1 Il catalogo metadati della Pubblica Amministrazione Locale piemontese

Il catalogo metadati della PAL piemontese nasce nel 1999, col nome di Infodir (information directory), e come evoluzione di precedenti esperienze di catalogazione metadati. Tale catalogo contiene e classifica inizialmente i metadati di business dei sistemi decisionali della Regione Piemonte. Nel 2002 cresce e progredisce secondo 3 principali dimensioni:

- enti e loro metadati censiti (non più solo Regione Piemonte, ma dapprima Città di Torino e successivamente Provincia di Torino, e via via i vari enti consorziati, ivi comprese le ASL e il sistema informativo interno del CSI stesso, che ha in carico la gestione dei metadati dei vari enti che popolano il catalogo)
- tipologia di servizi e basi dati censite, non più solo i sistemi decisionali ma anche gli operazionali
- granularità dei metadati, ovvero introduzione dei metadati tecnici delle basi dati (tavole, campi) e dei servizi applicativi (componenti architetture)

L'oggetto cardine del catalogo metadati è la collezione, intesa come tralcio del grappolo di metadati ad essa associato, costituito da, basi dati, tavole, campi, applicativo, componente.

Ogni singolo oggetto è corredato da un set di metadati standard Dublin Core, tra i quali le descrizioni degli oggetti che sono la base per ricerche libere e guidate da criteri di ricerca testuale per somiglianza.

L'oggetto cardine, la collezione, è classificabile principalmente:

- per ente proprietario
- per materia della PA
- per tematismo trasversale a più materie ed enti

Le classificazioni consentono il browsing, le ricerche libere il searching.

Browsing e searching sono completamente indipendenti e non combinabili tra loro.

Le modalità di searching possono essere sia top-down (dalla collezione all'attributo del data base) che bottom-up (dal singolo campo del data base a risalire fino alle collezioni e alle classificazioni che le contengono).

I metadati tecnici censiti delle varie basi dati sono perlopiù fotografie degli schemi logico-fisici delle basi dati stesse.

Nel 2004, per valorizzare il patrimonio di metadati censiti, è stato sperimentato un metodo ed un tool a corredo di Infodir, che consentisse il raggiungimento dei seguenti principali obiettivi: addivenire ad un embrione di tassonomia dinamica, corredata da criteri like, per raggiungere i metadati per somiglianza dei nomi degli elementi delle tassonomie stesse; consentire un mutuo scambio di inferenza tra le tassonomie utilizzate, in realtà costituite dagli schemi concettuali della Pubblica Amministrazione centrale, e i constraints presenti nelle strutture delle basi dati logiche censite.

Per fare questo si sono implementate le tassonomie, intese come gerarchie di entità, afferenti alle entità principali della PA centrale:

- soggetto
 - fisico (persona fisica)
 - * lavoratore
 - autonomo
 - dipendente privato
 - dipendente pubblico
 - * imprenditore
 - giuridico (imprese)
- cosa
 - bene
 - documento
- geografia
 - luogo
 - territorio,
 - urbanistica

Ciascun livello delle singole tassonomie ha associato un criterio di somiglianza che pesca dai metadati descrittivi tecnici delle componenti delle basi dati (tavole, campi). Si sfruttano inoltre le relazioni tra le tassonomie/gerarchie, ad esempio: *cittadino paga tributo* (cittadino elemento della gerarchia soggetto fisico, tributo elemento della gerarchia bene) per inferire dall'alto al basso relazioni tra gli oggetti censiti.

Mutuamente, gli oggetti logico-fisici censiti delle basi dati, avendo tra di loro dei constraints, forniscono inferenza dal basso all'alto, quindi relazioni, tra gli elementi delle tassonomie/gerarchie. Questa tecnica di inferenza tassonomico-ontologica attuata su Infodir, consente, per ogni singolo concetto della PA, di verificare in quali basi dati questo concetto è fisicamente istanziato, come è correlato o correlabile sia top-down che bottom-up.

Il tool sperimentale realizzato, consente, data una singola struttura di base dati, di inferire tutti i possibili concetti presenti per somiglianza nei metadati descrittivi della base dati stessa. Tali criteri sono generalizzabili a qualsiasi oggetto di portale, classificato con qualsiasi tassonomia.

Ad inizio 2007 è stata rilasciata una nuova versione di Information Directory che supera alcuni limiti architetture del vecchio Infodir nato nel 1999. Esso è stato infatti generalizzato e potenziato, nonchè condiviso fra le 3 principali pubbliche amministrazioni piemontesi (Regione, Città e Provincia di Torino), e riconosciuto come il nuovo Infodir. Il nuovo Infodir annovera tra le sue principali caratteristiche le seguenti:

- il backend decentrato presso i centri di competenza per materia presenti in CSI e presso gli enti
- viste separate e viste condivise dei metadati, sia di business che tecnici
- oggetti generalizzati
- classificazioni dinamiche, ovvero tassonomie, generalizzate, ed associabili a criteri di text mining che permettono di classificare automaticamente gli oggetti via via censiti

4.2.2 Esperimenti di estrazione e correlazione di concetti

In questo abstract si intende descrivere come estrarre e derivare conoscenza dai metadati descrittivi degli oggetti dei portali; come utilizzare criteri di somiglianza per l'estrazione della conoscenza.

Per fare questo si sono implementate le tassonomie sopra descritte, intese come gerarchie di entità, afferenti alle entità principali della PA centrale.

Ad ogni elemento di dettaglio delle tassonomie, delle quali per brevità abbiamo sopra evidenziato solo la struttura di livello concettuale più alto, sono stati correlati opportuni criteri di somiglianza da applicare, nel retrieval, ai metadati descrittivi degli oggetti fisici delle basi dati (tavole, campi). In tal modo abbiamo reso possibile verificare, dato un concetto, in quali strutture dati il concetto è presente per somiglianza.

Successivamente abbiamo sperimentato la strada inversa, ovvero data una struttura dati, risalire ai concetti che essa contiene, sempre per somiglianza.

Nel 2004, per valorizzare il patrimonio di metadati censiti, è stata realizzata insieme al professor Batini dell'Università Bicocca di Milano una metodologia e un tool sperimentale a corredo del catalogo metadati, che consentisse il raggiungimento dei seguenti principali obiettivi: - addivenire ad un embrione di

tassonomia dinamica, corredata da criteri like, per raggiungere i metadati per somiglianza dei nomi degli elementi delle tassonomie stesse - consentire un mutuo scambio di inferenza tra le tassonomie utilizzate, in realtà costituite dagli schemi concettuali della Pubblica Amministrazione centrale, e i constraints presenti nelle strutture delle basi dati logiche censite. Dopo questa esperienza, ci siamo posti alcune riflessioni sul come generalizzarla. Poichè il focus è sui metadati descrittivi (nomi, descrizioni) ai quali si applicano i criteri di ricerca, cosa c'è di diverso tra:

- una struttura dati (tavola, campo)
- un servizio descritto in un portale (servizio, componente architetturale)
- altro oggetto, ad esempio un filmato

In altre parole, è possibile sofisticare i criteri (ad esempio usando il text mining) per far sì che - incrementando la base di conoscenza concettuale - facendo lavorare la base di conoscenza e i criteri su qualsiasi oggetto di portale si giunga a quella che è la nostra interpretazione di web semantico?

Esperienze fatte con il tool realizzato con l'università Bicocca di Milano

Tale metodologia è stata messa a punto con la collaborazione del professor Batini, tool realizzato in tesi di informatica con il dottor Manuel Francesco Garasi. Il tool è scritto in java.net, database access, desktop client. è in corso di stesura presso Bicocca la versione Java/Mysql/web. è un repository di schemi concettuali della PA, italiana e locale, visitabile ad albero ed organizzato con algoritmi di integrazione/astrazione schemi. Gli schemi della PA centrale sono stati resi in forma elettronica partendo dai lavori degli anni '90 del professor Batini. Gli schemi della PA locale piemontese sono ottenuti con la metodologia Batini-Grosso basata su criteri di somiglianza ed inferenza. Il tool consente di visitare gli schemi base e la piramide di schemi integrati e astratti. In modalità PAC (PA centrale) gli schemi sono visitabili as is, in modalità PAL (PA locale) sono ottenibili mediante riconcettualizzazione dal fisico. Per ogni schema, PAC o PAL, il tool attiva erwin che procede con il disegno dello schema stesso. è stato scelto erwin come tool grafico, anche se è penalizzante perchè non ha tutti i costrutti tipici della modellazione dati concettuale, ovvero le gerarchie di generalizzazione, e nemmeno la possibilità di avere le relazioni espresse come oggetti, non solo come linee corredate da cardinalità. A questo si è ovviato usando le stesse entità e prefissando le generalizzazioni con G- e le relazioni con R- Le entità di livello più basso nelle gerarchie hanno invece il prefisso E- Il vantaggio di erwin è che, nelle release più recenti (dalla 4 in poi) consentono l'export delle strutture dati in formato XML. Il tool, tra le funzioni di ricerca delle quali è dotato, consente la ricerca per nome anche parziale delle entità, fornendo:

- per gli schemi base una lista di impact analysis che contiene i nomi degli schemi che racchiudono una certa entità

- per gli schemi astratti l'evidenza in rosso, nelle piramidi di schemi, degli schemi astratti che racchiudono una certa entità

Per ogni schema è possibile vedere le seguenti componenti concettuali:

1. entità
2. attributi
3. generalizzazioni
4. relazioni
5. constraints

Dallo schema fisico è possibile vedere una o più componenti concettuali, e combinazioni di esse: 12, 13, 14, 15, 123, 124, 125, 134, 135, 145, 1234, 1235, 1245, 1345, 12345.

È possibile riconcettualizzare uno schema fisico a partire dalle seguenti piattaforme: Access2000, MySQL, Oracle. È corredato da un help online organizzato secondo 3 principali paragrafi: introduzione, modellazione concettuale, uso del repository è possibile arricchire il database con:

- nuove gerarchie di generalizzazione
- nuovi schemi

Non è possibile, allo stato attuale, creare una nuova base di conoscenza (ad esempio PA-egov, oltre a PAC e PAL già previste). I punti di forza del tool, che merita investimenti per generalizzarlo al web, sono i seguenti:

- import gerarchie di generalizzazione
- versione web in corso di realizzazione
- tecnologia java/mysql (open source oriented)

L'idea che fa seguito al lavoro svolto sinora con i metodi e i tool descritti, consentirebbe di fornire ai power-user degli enti PA piemontesi uno strumento basato su mappe concettuali.

L'ipotesi di fondo è quella di generalizzare i metodi e i tool descritti, realizzando un prodotto logico. Tale prodotto logico deve avere una interfaccia grafica simile a quella di tool come ad esempio CMAP, e deve essere costituito da una parte alta che contiene librerie di schemi concettuali (ad esempio della PA) e da una parte bassa costituita da oggetti censiti nei portali (ad esempio oggetti della PA, di varia tipologia, basi dati, servizi, prodotti, filmati, etc.).

Ciò è generalizzabile, ovvero, coerentemente con la materia della quale si vuole fornire una interfaccia semantico-ontologica, è possibile ad esempio: inserire in parte alta una libreria di schemi concettuali di sistemi gestionali; inserire

in parte bassa i metadati degli oggetti di un portale di e-commerce. Il prodotto deve essere in grado di inferire conoscenza, con le tecniche sperimentate e descritte prima, sia bottom-up sia top-down.

Con queste premesse, ecco quali sono i punti deboli, da rinforzare, dei tool sviluppati: il tool grafico attivabile, cioè erwin, che anche se consente export xml di strutture dati e metadati non è adatto al web, non consente entità cliccabili con ipertesti; meglio altre realizzazioni quali <http://cmap.ihmc.us/>. Inoltre, se i metadati sono poco descritti, la riconcettualizzazione per somiglianza è debole.

È possibile attuare alcune leve migliorative:

- migliorare la qualità dei metadati aumentando le descrizioni
- aumentare la base di conoscenza concettuale del tool, la sua intelligenza, la precisione del retrieval sia sui nomi che sulle descrizioni
- consentire, oltre alla funzione già esistente di import nuove gerarchie di generalizzazione, anche import di nuova base di conoscenza, ad esempio PA-egov, import schemi, import entità, import attributi e import relazioni

A questo punto, se la base di conoscenza alta utilizzata per l'inferenza dei concetti è generalizzabile, è possibile fare riuso anche della parte bassa, ovvero i metadati descrittivi, che oltre a tavole e campi possono essere qualsiasi oggetto censito in un portale, purchè corredato da metadati descrittivi Dublin Core. Così come i criteri di retrieval agiscono su:

- nomi tavole
- descrizioni tavole
- nomi campi
- descrizioni campi

generalizzando possiamo far agire i criteri su:

- nomi oggetti padre
- descrizioni oggetti padre
- nomi oggetti figlio
- descrizioni oggetti figlio

Padre e figlio possono essere oggetti metadatati e correlati censiti in un portale, ad esempio servizi e prodotti della PA. Si vorrebbe quindi sperimentare la strada della generalizzazione dei tool Bicocca (librerie di schemi concettuali usate per le inferenze sui metadati descrittivi degli oggetti dei portali): - sofisticando i criteri di retrieval con metodi e tools di text mining:

- accorciando se necessario le stringhe di retrieval per migliorare il retrieval stesso sui nomi degli oggetti, ove mancano le descrizioni

- arricchendo le gerarchie di concetti, e le relazioni, anche al di sotto del valore soglia, fino ad utilizzare per il retrieval le entità (e i criteri di retrieval associati) degli schemi base PA
- arricchendo le gerarchie di concetti, e le relazioni, con gli schemi PA-egov di Arianna-Diviana.

4.3 Sviluppo di un'ontologia per l'eGovernment: un caso di studio italiano

Basato su materiale contribuito da:

Flavio Corradini, Francesco De Angelis, Alberto Polzonetti, Elia Brugnoli
Dipartimento di Matematica e Informatica, Università di Camerino

4.3.1 Abstract

Il concetto di interoperabilità diviene oggi centrale per un reale piano di attuazione dell'eGovernment. Allo stesso tempo la tendenza è quella di evolvere verso il livello semantico prevedendo l'uso di tecnologie come le ontologie per la rappresentazione della realtà inerente il dominio in cui i soggetti operano. In questo lavoro si presenta un'ontologia per uno dei più complessi documenti della Pubblica Amministrazione che potrebbe porsi alla base per la realizzazione di software in grado di aumentare l'efficienza nella collaborazione tra le varie PA. Ciò permette la semplificazione dei rapporti che sussistono tra cittadini e uffici pubblici consentendo inoltre anche un perfezionamento dei servizi offerti dalla Pubblica Amministrazione come la redazione di documenti o lo storage delle informazioni correlate.

4.3.2 Introduzione

Nei sistemi di eGovernment è di fondamentale importanza garantire l'interoperabilità e la cooperazione applicativa tra sistemi informativi delle PA. Ciò risponde a due esigenze fondamentali di realizzazione e successivo sviluppo dell'eGovernment:

- Integrazione dei processi automatizzati di back-office per l'erogazione di servizi;
- Erogazione di servizi finali al cittadino in modo trasparente tramite un unico punto di accesso.

Per far fronte a queste esigenze, le PA devono far evolvere i loro sistemi informativi con un approccio cooperativo e basato su di una conoscenza condivisa tra le parti. Ciò comporta la necessità di realizzare delle infrastrutture che permettano ai sistemi ed applicazioni di e-government di interoperare, secondo le esigenze dei vari domini applicativi.

Con i termini *interoperabilità* e *cooperazione applicativa* si fa riferimento alla capacità di un sistema informativo di disporre automaticamente, per le proprie finalità, di dati che sono acquisibili solo tramite un altro sistema informativo affinché le applicazioni di cui è costituito siano in grado di operare. In particolare, l'*interoperabilità* fa riferimento alla capacità di due o più sistemi informativi di scambiarsi informazioni e di attivare a questo scopo processi nelle rispettive applicazioni. Un possibile approccio è, ad esempio, l'adozione di uno stesso formato di interscambio dei dati e di un protocollo di comunicazione condiviso.

La *cooperazione applicativa*, inoltre, si pone ad un livello superiore indicando la capacità di uno o più sistemi informativi di avvalersi, ciascuno utilizzando la propria logica applicativa, dell'interscambio automatico di informazioni con gli altri sistemi ottenuto con l'interoperabilità. Quest'ultima è, quindi, un prerequisito essenziale per la *cooperazione applicativa*. Il concetto di interoperabilità esteso a livello semantico [6] rappresenta la capacità dinamica derivante dall'applicazione di tecnologie software (come ontologie, motori di inferenza, reasoner) in grado di inferire, mettere in relazione, interpretare e classificare il significato implicito dei dati (o in generale dei contenuti digitali). Tramite l'uso di queste tecnologie è possibile rappresentare in maniera efficiente ed efficace la realtà inerente il dominio in cui le applicazioni lavorano. Ciò contribuisce in maniera determinante al miglioramento dello scambio di informazioni tra diversi soggetti che cooperano per il raggiungimento di uno specifico obiettivo. L'interoperabilità semantica condivide strumenti e metodologie proprie del Semantic Web ma si prefigge di raggiungere un sottoinsieme dei suoi obiettivi. Mentre il Semantic Web [9] focalizza sullo scambio di informazioni e sulla comprensibilità del loro significato in maniera autonoma e globale sul web, l'interoperabilità semantica mira ad arbitrare e mediare strutture, contesti e significati relativamente a domini conosciuti i cui confini sono ben delineati. L'interoperabilità a livello semantico si fonda su di una rappresentazione della realtà in grado di descrivere il dominio di riferimento con metodi che siano poi utilizzabili dalle applicazioni che fanno uso di tale rappresentazione. Linguaggi adatti a tali descrizioni sono basati su un qualche formalismo logico che consente un determinato livello di espressività. Inoltre, il linguaggio utilizzato deve poter essere facilmente utilizzabile sul web. Attualmente OWL (Ontology Web Language) [10] è il linguaggio XML-based maggiormente utilizzato per le sue caratteristiche di espressività poiché si basa sulla Description Logic [5] che fornisce un sottoinsieme della logica dei predicati del primo ordine. E' naturale pensare che con questa evoluzione del concetto di interoperabilità anche nel dominio dell'eGovernment occorrerà avere un metodo per la rappresentazione formale della realtà di interesse. Ciò permette la semplificazione dei rapporti che sussistono tra cittadini e uffici pubblici consentendo inoltre anche un perfezionamento dei servizi offerti dalla Pubblica Amministrazione come la redazione di documenti e lo storage delle informazioni. Il resto del lavoro è organizzato come segue. La Sezione 2 descrive il documento "Atto di Nascita" così come previsto dalle normative italiane. La Sezione 3 illustra l'ontologia di riferimento realizzata mentre la Sezione 4 conclude il lavoro.

4.3.3 Il documento "Atto di Nascita"

Il documento preso in esame in questo articolo è l'Atto di Nascita. L'iter amministrativo inizia con la dichiarazione di nascita [1, 2, 3, 7] che è effettuata da uno dei genitori o da un loro sostituto rispettando l'eventuale volontà della madre di non essere nominata. Il rilascio del documento può essere effettuato entro dieci giorni dalla nascita presso il comune nel cui territorio è avvenuto il parto o in alternativa, entro tre giorni, presso la direzione sanitaria dell'ospedale dove è avvenuta la nascita. In quest'ultimo caso, la dichiarazione può con-

tenere
anche il riconoscimento contestuale del figlio naturale. Ai fini della successiva trascrizione, la dichiarazione è trasmessa all'Ufficio di Stato Civile del comune.

Le entità della PA interessate all'iter dell'atto di nascita sono:

- Direzione Sanitaria dell'Ospedale o della Casa di Cura ove è avvenuta la nascita
- Comune di residenza della madre, se il padre risiede in altro Comune
- Comune ove è avvenuto il parto -Comune di residenza dei genitori
- Comune di residenza del padre, previo accordo con la madre, residente in altro Comune

Accolta la dichiarazione e la documentazione, l'Ufficiale di Stato Civile redige l'atto di nascita dandone comunicazione all'Ufficio anagrafe affinché provveda al rilascio del tesserino di Codice Fiscale del bambino e del certificato di Stato di famiglia che i genitori dovranno esibire all'A.S.L. per l'acquisizione del libretto sanitario e per la scelta del Pediatra. Da quanto appena esposto si deduce come la realizzazione ed rilascio dell'Atto di nascita sia un evento complesso che richiede la cooperazione tra diversi soggetti dalla PA. Su tale documento si basa il caso di studio in esame analizzato nella Sezione 3.

4.3.4 Descrizione dell'ontologia

Struttura dell'Atto

In questa sezione si descrivono le varie componenti del documento relativo all'“Atto di Nascita” e la sua rappresentazione ontologica. La struttura di questo documento amministrativo [4] può variare a seconda delle diverse circostanze che possono manifestarsi al momento della dichiarazione di nascita. Conseguenza di ciò le norme prevedono ben tre tipi di atto che vengono identificati rispettivamente con le sigle 1A, 1Be 2A. Al fine di analizzare in maniera completa le differenze che sussistono tra le diverse tipologie si effettua una suddivisione generale del documento identificando per ognuna di esse una parte iniziale, una centrale ed una finale. La prima parte è quella che identifica l'intestazione del documento ed è comune negli atti 1A e 1B ma diversa in 2A. La parte centrale rappresenta il corpo dell'atto la cui struttura è uguale per tutte le tipologie pur contenendo differenti clausole testuali. L'ultima parte, conclusiva del documento, è indipendente dalle scelte iniziali. La struttura dell'atto, visibile in Figura 1, richiede ulteriori raffinamenti sulla rappresentazione che permetta di capire come le clausole in gioco varino durante la dichiarazione di nascita. Come risultato dell'analisi effettuata è emerso che l'atto 1A rappresenta la base su cui le altre tipologie si appoggiano. Per descrivere questo importante aspetto viene definito nell'ontologia in concetto *Atto_1A* che dispone delle sottoclasse per la rappresentazione degli atti 1B e 2A. La classe *Atto_1A* fornisce, oltre a

varie clausole che verranno descritte successivamente, degli identificativi propri di ogni istanza dell'atto. In particolare si hanno un numero identificativo ed il cognome, nome e sesso del bambino a cui l'atto si riferisce. Per modellare questi aspetti di carattere generale sono stati creati un attributo ed una relazione. Il codice identificativo è chiaramente un attributo dell'atto mentre per le informazioni riguardanti il nascituro è stata creata una relazione con il concetto *Nascita* poiché l'atto ufficializza un evento di questo tipo.

L'evento "Nascita", modellato nell'ontologia con l'omonimo concetto, può essere caratterizzato da una serie di particolarità e caratteristiche che devono essere riportate nell'atto: La nascita può essere *legittima, naturale riconosciuta* o *non riconosciuta* e come è facile intuire queste tre possibilità sono stati espresse come delle sottoclassi di *Nascita*.

La relazione "è_Riferita" definisce come codominio la classe *Bambino*, sottoclasse di *Persona*. Questa classe eredita dalla classe *Padre* tutte le informazioni inerenti una persona: il nome, il cognome, il sesso, il luogo di nascita, etc.

Altri dati relativi ad un individuo sono espressi con delle relazioni che hanno come dominio la classe *Persona*. La classe *Bambino*, in Figure 4.2 e 4.3, rappresenta il dominio di quattro fondamentali relazioni:

- *Clausola di Presenza*: collega la classe *Bambino* ad una sottoclasse della classe *Clausola* ed indica la presenza fisica del bambino al momento della stesura dell'atto
- *Possiede*: collega il bambino alla classe *Dati_Di_Nascita* che fornisce informazioni come la data e l'orario di nascita del bambino. Inoltre tramite la proprietà "è_Dichiarazione_Di" è possibile associare questi dati alla classe *Atto_1A* poiché queste informazioni nel documento cartaceo dell'atto rappresentano la dichiarazione di nascita vera e propria.
- *Viene_Presentato_Da*: attesta la presenza di un presentatore al momento della stesura dell'atto. La classe che descrive il presentatore risulta essere sottoclasse di *Persona* permettendo l'ereditarietà dei dati visti in precedenza. Il ruolo del presentatore del bambino può essere assunto da uno dei due genitori o da un loro procuratore, dal medico o dall'ostetrica. Per questo sono state create nell'ontologia quattro sottoclassi di *Presentatore* ad indicare le diverse tipologie appena introdotte. Per quanto riguarda la classe *Genitore* è stata effettuata una ulteriore classificazione che consente di rappresentare i tre ruoli che possono essere ricoperti: *Padre, Madre* o *Vedova*. Inoltre, per quanto riguarda la classe *Procuratore* è stata inserita una relazione con il concetto di *Genitore* che consente di individuare il genitore che rappresenta (procuratore del padre, della madre, o della vedova). Al manifestarsi di una nascita del tipo *Naturale Riconosciuta* può agire da presentatore un genitore o un suo procuratore: il presentatore può indicare il nome dell'altro genitore, può essere accompagnato da esso/a o può non indicare alcun nome. Naturalmente la clausole del certificato di nascita subiscono cambiamenti sia in base alla scelta del presentatore che

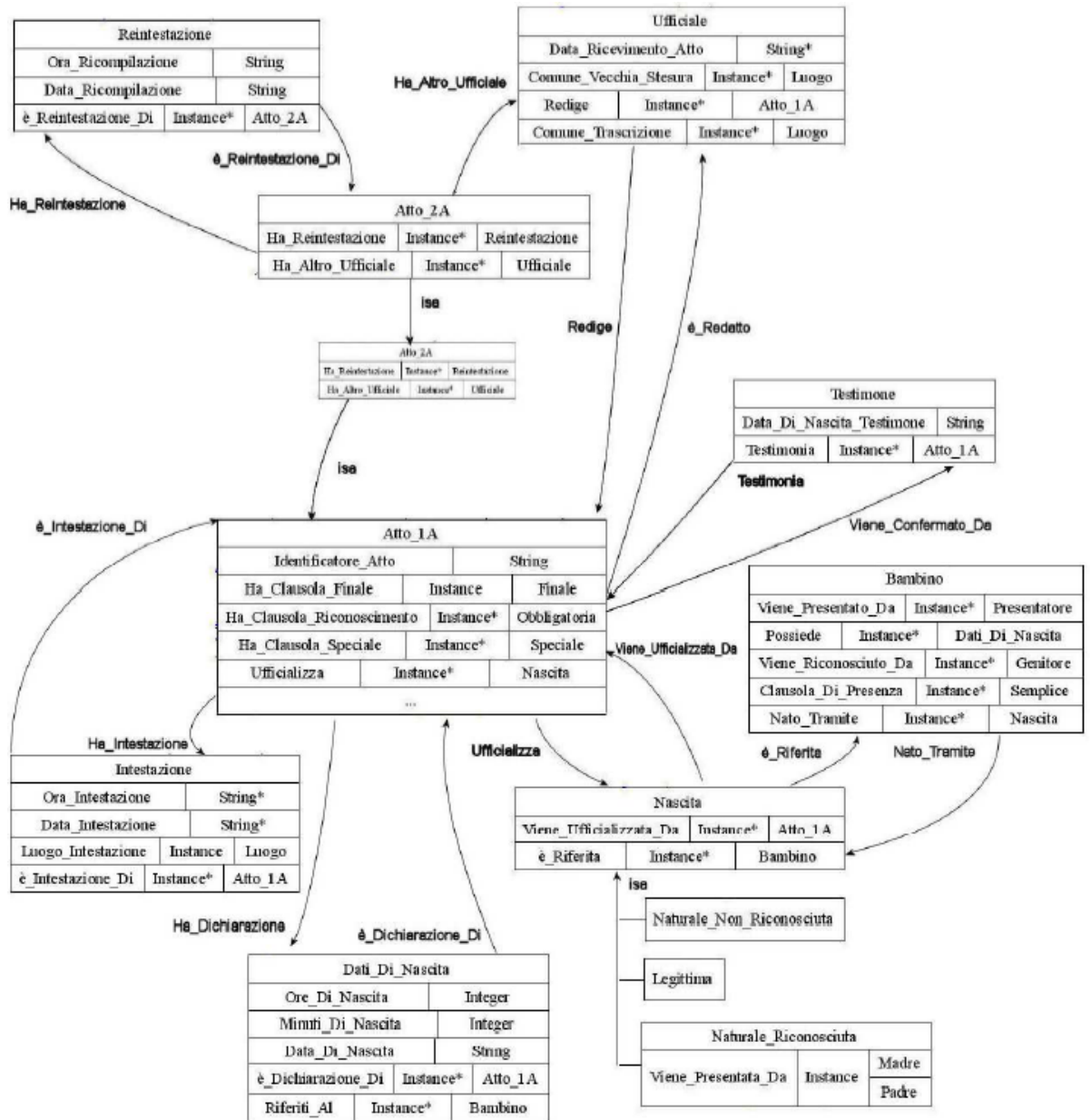


Figura 4.1: Struttura dell'Atto

alla sua tipologia (genitore o procuratore). Inoltre, visto che il presentatore può indicare lo stato dell'altro genitore si è ritenuto opportuno inserire nell'ontologia una relazione avente come dominio le classi *Padre* e *Madre* e come codominio la classe *Modalità_Di_Presentazione* che definisce tramite le sue sottoclassi lo stato dei due genitori.

- *Viene_Riconosciuto_Da*: permette, infine, di definire quale genitore riconosce il bambino. Come codominio ha la classe *Genitore* analizzata precedentemente discutendo il concetto di *Presentatore*.

Analizzando ulteriormente l'atto si possono introdurre altre tre importanti relazioni con dominio "*Atto_1A*" che permettono di descrivere altri aspetti del documento. La prima, "Ha_Intestazione", fornisce delle informazioni che riguardano la stesura dell'atto, rispettivamente: la data, l'orario di compilazione e il luogo. Per quel che riguarda quest'ultimo dato si utilizza una relazione che permette di collegare la classe corrente con la classe *Luogo*.

La seconda relazione, "Viene_Confermato_Da", ha come codominio la classe *Testimone*.

L'ultima relazione, non per questo la meno importante, "è_Redatto", collega la classe *Atto_1A* con la classe *Ufficiale*, sottoclasse di *Persona*, nella quale vengono definite informazioni che riguardano l'Ufficiale di Stato Civile responsabile dell'Ufficio. Inoltre è richiesto che l'ufficiale afferisca ad un comune identificato tramite una sede, una città ed una provincia. Quindi in questo contesto il comune rappresenta un luogo e tale aspetto è rappresentato tramite la relazione "Comune_Trascrizione_è". Infine è da notare come nel documento si richiedano informazioni sul nascituro come il nome, il cognome, il luogo di nascita, di residenza ed infine la data di nascita. E' intuibile che tutti i dati tranne la data di nascita possono essere ereditati dalla classe *Persona*.

Per la data di nascita è stata inserita una apposita relazione verso il concetto *Dati_Di_Nascita* in modo da raggruppare i dati caratterizzanti il documento "Atto di nascita".

Differenze tra i tipi di atto

Come descritto in precedenza l'intestazione dell'atto 2A si discosta dagli atti 1B e 1A poiché al suo interno vengono inserite informazioni supplementari che riguardano una nuova riformulazione dell'atto originale. Poiché queste informazioni riguardano anche l'ufficiale che compila il nuovo documento, sono state aggiunte alla classe *Ufficiale* due ulteriori proprietà che definiscono il comune della vecchia trascrizione e la data del ricevimento del vecchio atto. Un nuovo blocco di informazioni che devono essere aggiunte per formulare l'atto 2A riguardano l'orario e la data di compilazione del nuovo atto. Nell'ontologia questi dati sono individuabili all'interno della classe *Reintestazione* che risulta essere codominio della relazione "Ha_Reintestazione" nella classe *Atto_2A*. Ricordiamo che questa classe possiede solo queste due relazioni poiché è definita come sottoclasse dell'*Atto_1B* e naturalmente tutti gli attributi e tutte le proprietà definite

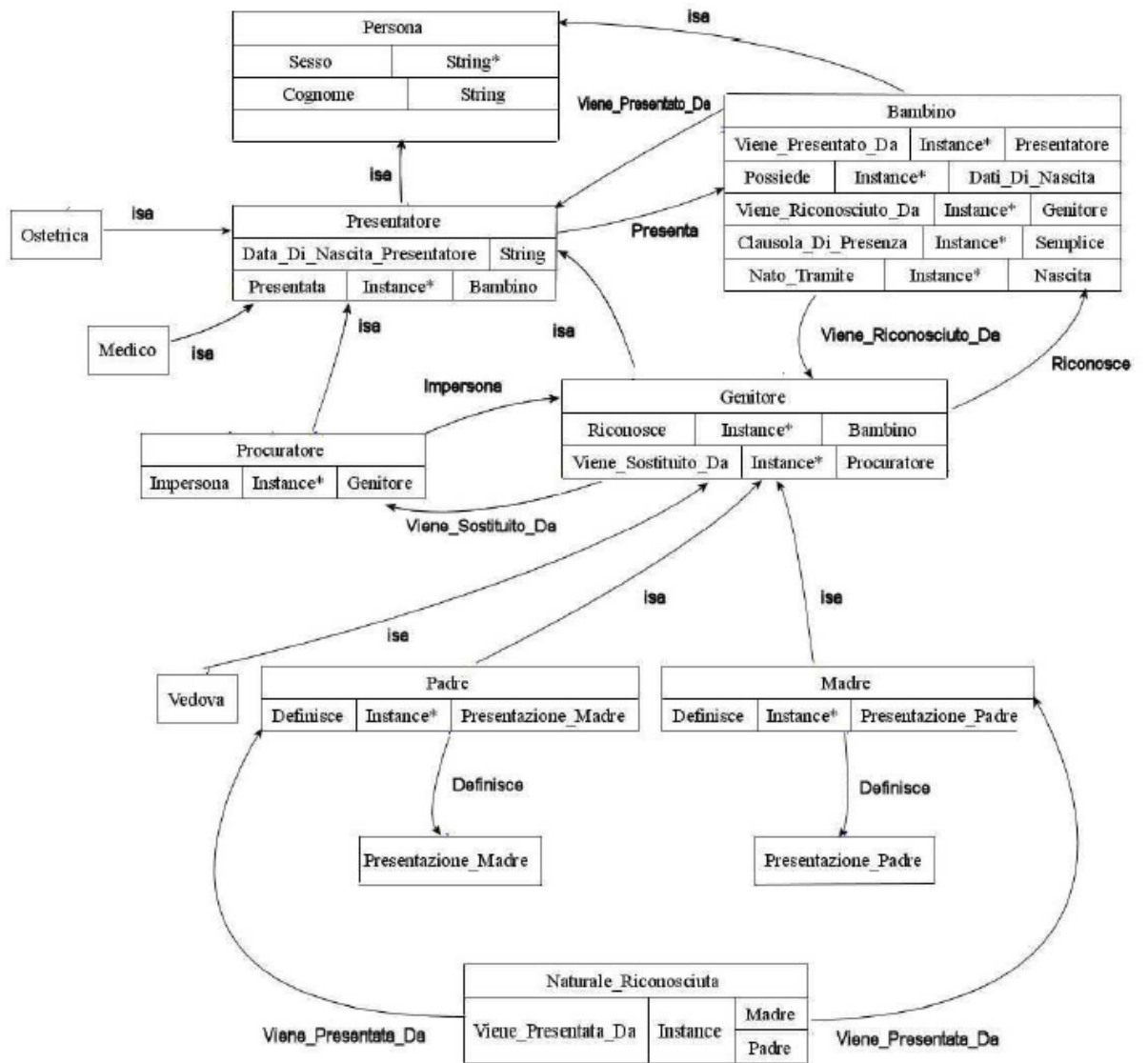


Figura 4.2: Concetto e relazioni del concetto Bambino

sopra per la creazione del documento vengono ereditate poiché *Atto_1B* è sottoclasse di *Atto_1A*. Il documento dell'atto 1B possiede, in più rispetto all'atto 1A, soltanto una sezione dove l'Ufficiale inserisce del testo libero per giustificare una dichiarazione tardiva. Nello schema ontologico è stata inserita la classe *Atto_1B* come sottoclasse di *Atto_1A* con l'aggiunta di un attributo che permette di descrivere questo aspetto.

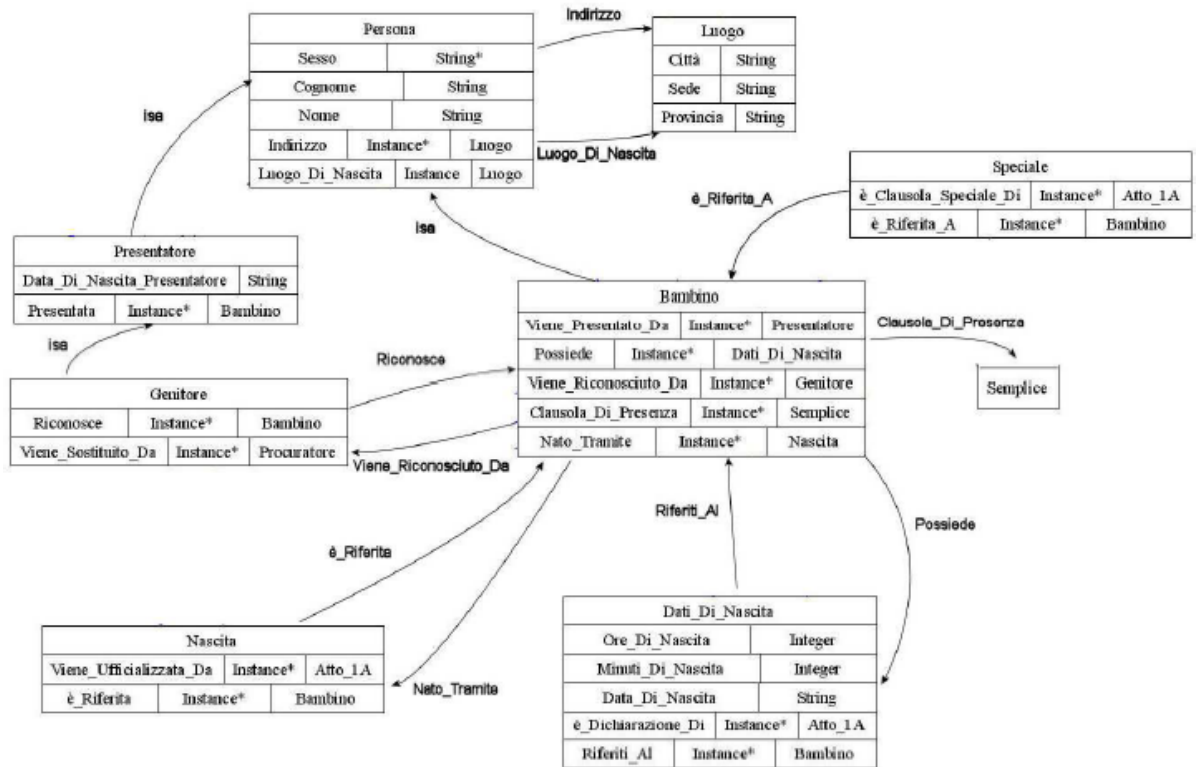


Figura 4.3: Altre relazioni del concetto Bambino

4.3.5 Descrizione delle Clausole

All'interno del documento esiste il concetto di *clausola* ossia di porzione di testo che svolge una determinata funzione all'interno dell'atto asserendo fatti che descrivono l'evento nascita. Le clausole presenti nel documento, indicate in Figura 4, vengono rappresentate da una classe dotata di due attributi: un identificativo ed un testo. Come si può intuire dai nomi, il primo identifica univocamente la clausola mentre il secondo permette di rappresentare il testo che la compone. Inoltre, le clausole possono essere *obbligatorie* oppure *conclusive*. La clausola obbligatoria può essere specializzata in *semplice*, *complessa* o *finale*. Per *complessa* si intendono tutte quelle clausole che sono dipendenti

dal tipo di presentatore o dalla nascita del bambino, mentre con *semplice* si fa riferimento a del testo semplice che viene inserito al momento della stesura dell'atto. Per quanto riguarda la classe che rappresenta le clausole finali, denominate *conclusive*, occorre precisare che esse concludono il documento e si riferiscono l'una al bambino e l'altra a chi sottoscrive l'atto. A sua volta la clausola di sottoscrizione del documento è anche sottoclasse della classe *Obbligatoria*. La prima clausola conclusiva indicata dalla classe *Speciale*, si scompone in tante sottoclassi quanti sono le diverse circostanze di un parto. A tale classe è stata associata la relazione "è_Riferita_A" per indicare che tale tipologia di classe è riferita ad un Bambino. Allo stesso modo la clausola che indica chi sottoscrive l'atto è stata scomposta in tante sottoclassi quanti sono i vari sottoscrittori. Le relazioni "Ha_Clausola_Special è'" e "Ha_Clausola_Final è'" della classe *Atto_1A* permettono di far riferimento rispettivamente a clausole *speciali* e *finali*.

Una clausola speciale può essere presente o meno nel documento in base alla presenza o meno di circostanze particolari verificatesi alla nascita. Tale clausola può assumere possibili forme come: *parto bigemino*, *plurimo primo*, *bimbo deceduto* o *nato già morto*, etc. Se presenti tali possibilità non dipendono dal fatto che sia il padre, o la madre, o il procuratore a fare la dichiarazione. Tale scelta, quindi, può avvenire anche successivamente alla compilazione dell'atto di nascita.

Una clausola finale, al contrario della clausola speciale, è sempre presente ed è anch'essa indipendente dalla figura che presenta il bambino indicando, però, chi sottoscrive il documento. Il sottoscrittore può essere classificato in uno dei seguenti modi:

1. Forestiero;
2. Sordomuto;
3. Nessun sottoscrittore per impedimento da parte dei presenti;
4. Tutti i presenti;
5. Alcuni dei presenti;
6. Nessuno dei presenti;

4.3.6 Conclusioni

Da quanto precedentemente esposto è evidente come la costruzione di un'ontologia per un atto della Pubblica Amministrazione Italiana sia un compito non banale in quanto esistono numerose leggi e norme dello Stato Italiano che regolano in maniera rigida la realizzazione del documento. In particolare, nel caso in esame le difficoltà maggiori si sono presentate sia nella rappresentazione e suddivisione delle diverse clausole che compongono l'atto, sia nella descrizione delle varie tipologie di presentatore previste dalla legge. Superate le difficoltà di rappresentazione e stabilita in maniera univoca, e condivisa tra le PA, la

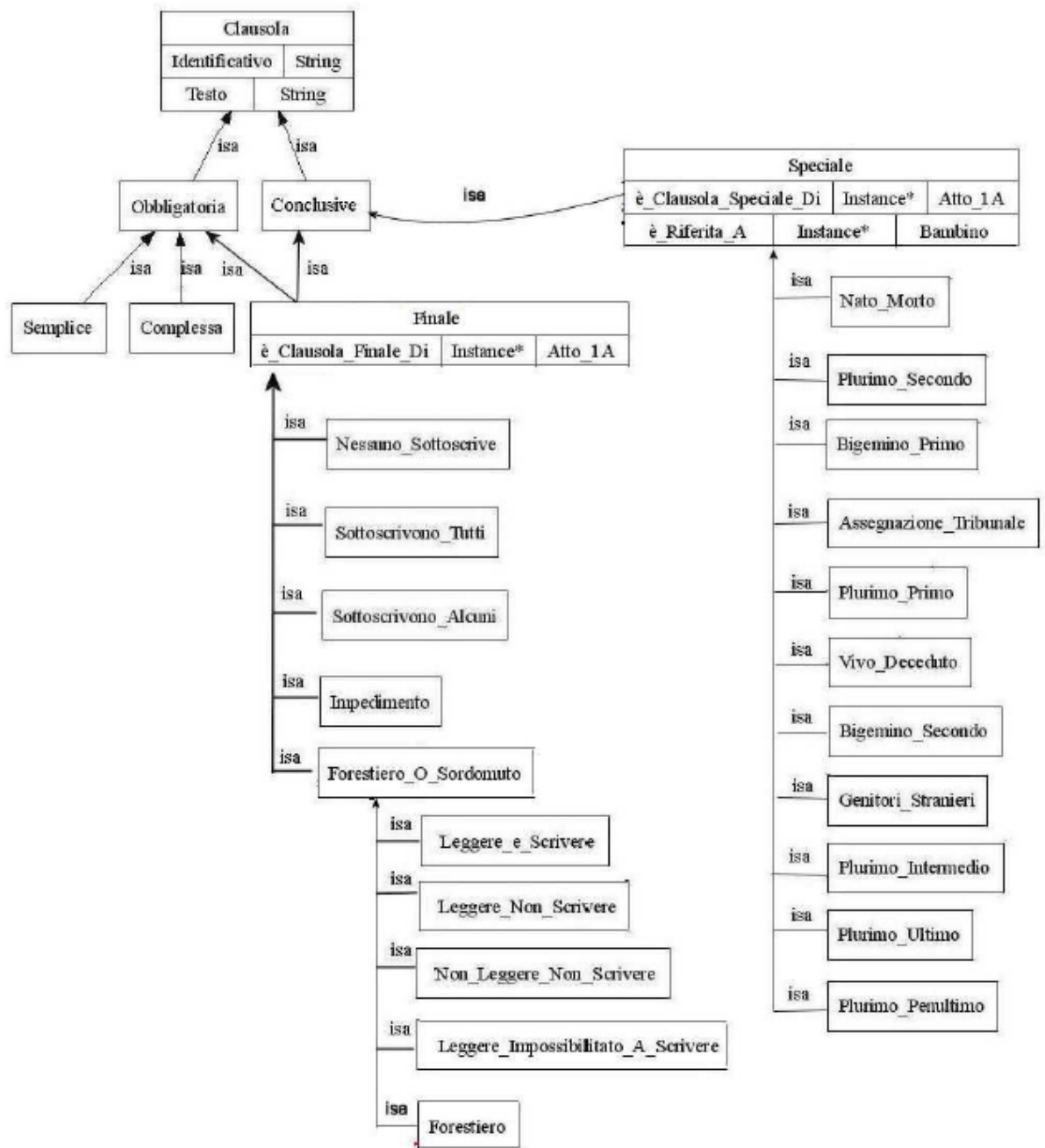


Figura 4.4: Rappresentazione delle clausole

rappresentazione dell'atto questa si pone come base per l'attuazione dell'interoperabilità a livello semantico tra le PA italiane che trattano documenti e dati inerenti le nascite. Tra le possibili applicazioni è utile segnalare che con l'ausilio di questa rappresentazione si fornisce la possibilità di realizzare un editing intelligente in grado da fornire una guida nella stesura dell'atto evitando errori ed incongruenze sia nel documento stesso sia nello storage delle informazioni in esso contenute. Queste ultime potranno essere maggiormente fruibili dalle varie PA grazie alla migliore espressività dei linguaggi utilizzati per la loro rappresentazione.

4.3.7 Riferimenti bibliografici

- [1] Codice Civile, artt 231 e segg., Dello stato di figlio legittimo
- [2] Diritto di famiglia, Legge 19/05/1975, n. 151
- [3] DPR 3.11.2000, n. 396, *Regolamento per la revisione e la semplificazione dell'ordinamento dello stato civile, a norma dell'art.2, c.12, della legge n.127/97*
- [4] Fabio Scarpantoni, 2004, *Soluzioni di eGovernment*, Tesi di Laurea in Informatica Università di Camerino
- [5] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Pater-Schneider, 2003. *The Description Logics Handbook: Theory, Implementation and Applications*. Cambridge University Press.
- [6] Jeffrey T. Pollock and Ralph Hodgson, 2004, *Adaptive Information*, Wiley
- [7] Legge 15.5.1997, n. 127, *Misure urgenti per lo snellimento dell'attività amministrativa e dei procedimenti di decisione e di controllo*
- [8] Steffen Staab, Rudi Studer, 2003, *Handbook on Ontologies*, Springer-Verlag
- [9] Tim Berners-Lee, James Hendler and Ora Lassila, 2001, *The Semantic Web*, Scientific American
- [10] W3C, *OWL Ontology Web Language*, <http://www.w3.org/TR/owl-guide/>

NOTA: Questo contributo deve ancora essere tradotto in italiano.

4.4 A case study of semantic solutions for citizen-centered Web portals in eGovernment: the Tecut Portal

Basato su materiale contribuito da:
Flavio Corradini, Alberto Polzonetti e Barbara Re Dipartimento di
Matematica ed Informatica, Università di Camerino, Italy

4.4.1 Abstract

Web portals are emerging as significant tools for eGovernment. Portals are the “gateways” between citizens and Public Administrations. Although a number of them have been already developed, shortcomings related to interoperability and usability limit their usage and potential. To improve their performance, we propose a semantic approach based on the so-called “Life Events”. This approach provides several advantages related to service automation and enhanced searching. Also, the usability offered to the end users is improved. To validate our techniques, the proposed approach has been applied to a real case study: the Tecut Portal.

4.4.2 Introduction

Web portals are playing an important role in the provision of digital services for citizens and Public Administrations, here after PAs. The evolution from the old-fashion Web sites to the current Web portals has allowed the development of new ways of doing business, learning, accessing services ... They are referenced, in the modern information society, as eTechnologies. At the same time, PAs noticed the emerging of Web portals as significant tools enabling eGovernment and they introduced them as gateways to interact with citizens. Of course, the introduction of Web portals allow the reduction of time and cost both for Public Administration and for citizens.

A number of eGovernment portals have been already developed even though, in several cases, shortcomings related to interoperability and usability limit their usage and potentiality. Due to the unavoidable need for service integration, interoperability concerns must be solved. This issue involves concerns at administrative, operational, technical, semantical, legal and cultural level. Thus, PAs must perform a long-term study to evaluate how to deploy their solutions. These ones must provide the highest possible level of satisfaction to really increase the level of interaction with citizens.

This paper intends to show the implementation of a solution offering customer-oriented services and the integration of the former in a Web portal. A semantic-based approach on the so-called “Life Events” is followed to drive proposed fea-

tures. Our proposal allows several advantages such as automatic services composition, advanced searching mechanisms, new functionalities as well as a better usability from the point of view. Summing up, our approach provides a more friendly users support for eGovernment services. To validate our techniques, a real case study has been developed: the Tecut Portal.

The rest of the paper is organized as follows. Firstly, we present the eGovernment state of the art. Secondly, we introduce the Tecut Portal, the study case we are dealing with. Next section introduces the concept of Life Event as it is going to be considered in the proposed framework. Later on, we introduce semantics in the system to model in a formal way the use of LE and to support its invocation. Finally, some future work and conclusions are yielded.

4.4.3 State of the Art

Web portals represent integrated gateways for government service between PA and users to provide a single point of contact for services. The goal of eGovernment services is to conduct complete end-to-end solutions for citizens whenever it is possible. As portals integrate services, they certainly improve access to government, reduce service-processing costs, and enable PAs to provide a higher quality of service.

The development of Web application for eGovernment services has benefits for both government and citizens. Allowing the access to information and services by means of Web interfaces, citizens and businesses can now access and interact with PAs under a 24/7 model.

We can also make a distinction related to the level of possible interaction in eGovernment solutions.

1. Emerging presence (stage I). Just information is presented and documents are available only for download.
2. Enhanced presence (stage II). Citizen can search for documents and perform more advanced operations; nevertheless, citizen can submit very little amount of information to PAs.
3. Interactive presence (stage III). Interactive services are available and government officials can be contacted by email, fax and telephone.
4. Transactional presence (stage IV). Two-way interaction is supported and complex services (such as taxes, fees and postal services) are available.
5. Networked presence (stage V). Final level that integrates all services under ICTs platforms and support a two-way open dialog between citizens and PAs.

The highest functioning Web portals show a complete system integration across agencies whereas portals with the lowest level of functionality provide little more than access to forms and static bits of information. High-functioning

portals create a true one-stop shop for citizens. In particular usability, customization, openness, and transparency represent the key aspects of portal functionalities.

As matter of fact, Web portals from PAs can take advantage of semantic solutions to solve issues related to organizational heterogeneity, interoperability and information accumulation. Information and services can be provided by different governmental agencies in different locations and the user does not necessarily know the organizational structure and who is responsible for each service. In these contexts, semantic is expected to play a relevant role. In literature we can find some interesting initiatives, at different levels, that make use of semantic:

1. e-GIF (eGovernmentInteroperability Framework) is the technical guidance for deploying eGovernment in the United Kingdom. Among many other national initiatives, it is relevant for our case because of they support for semantic features as in e-GSM (eGovernmentMetadata Standard).
2. Several projects supported by the EU through the Framework Program must be cited:
3. OntoGov. This project deals with the problem of services in eGovernment under a semantic point of view and it is aimed to provide an ontology to model the problem in a maintainable way.
4. Terregov. This project's main goal is to provide an interoperable layer that allows citizen to access eHealth services in a transparent manner by means of web services.
5. The SemanticGov project. This project aimed developing a software infrastructure to provide support for PAs by means of semantic. Currently, it is an ongoing project.
6. Suomi. The Finnish portal for eGovernment services that provides a taxonomy for the classification of services.
7. EIP.AT. A project developed in the University of Linz, Austria, that addresses integration problems and tries to solve them by means of semantic modeling.

4.4.4 The Tecut Portal

Several Italian Regions were suggested to develop eGovernment solutions aimed at increasing interactions between Public Administrations and citizen by means of ICTs infrastructures. In order to accomplish this high level goal, several issues related to key aspects in the eGovernment domain have to be taken into account, such as authentication and authorization, service publishing and discovery as well as composition. As results of these considerations and according to a recent study about skills for the case, it was developed the Tecut Portal (see Fig. 4.5),

a fully integrated government portal for shared and standardized services within the Marche Region.



Figura 4.5: Tecut Portal home page

This study case is aimed at supporting activities of small and medium enterprises. Besides, the adaptability due to changes on political, social and economic conditions is a leading feature in the system.

A global vision of the Marche Region comprising further financial arrangements and aggregations, enterprises, banks and citizens is provided. This clearly boosts the national and international chances to increase relations with PAs and drive advanced ways to improve standards of living. As a matter of fact, the Marche Region is among the first places in Italy as far as welfare, cohesion and competitiveness are concerning.

Even a lot of issues deserve a special attention, we would like to devote attention to a few of them. The authentication process plays a main role in Tecut. It represents the instant when the system determines the association between the digital identity and the user. The recent proliferation of digital services has raised concerns about a lot of authentication mechanisms. Marche Region supports the realization of a central authentication solution through Cohesion. It is an infrastructure that provides solutions for complex technical problems and a set of common standard services predisposed to realize applicative cooperation as the national eGovernment plan states.

Authentication services for centralized management access in private areas are provided by Single Sign On (SSO) and Profiling system.

1. The SSO's tasks are predisposed for the transfer of credentials between authenticated users and access portal. In particular, the authentication on the framework is possible with different levels: via weak registration using user-name and password and via strong registration using services regional cards "Raffaello". Furthermore, SSO allows a transparent access to the portal's reserved areas without further authentications and it allows that authentication credentials and user profiling are made available to different application domains. Indeed, the user authentication check is delegated to the service. It uses a regional services register to validate the profile in respect to the access roles.
2. The profiling system is dedicated to the coordinated management of credentials information, logically divided in a static subsystem and in a dynamic one, containing a series of attributes able to indicate the user's preferences when accessing the services. A part of user base profile will be requested during the registration phase, and another part is communicated after explicit request, when a service is used.

Processes related to discovery and composition of services were designed by means of LifeEvents, as explained later on. This new approach brings several advantages in the design and planning of solutions as shown in next sections.

Therefore, the portal is expected to offer a holistic support for on-line operations regarding Public Administrations within the Marche Region. Provided interfaces and information are expected to make easier citizen's life. At the same time the portal has become a reference point at organizational level providing back office governance. Currently, the portal is a gateway for 531 agencies, provides 65 different kinds of services and 34.515 digital services.

4.4.5 LifeEvents as organizer

The provision of advanced services and the so tight constrains related to interoperability lead us to the search of a common paradigm to build up facilities in an interoperable and effortless manner. From the study of the domain and the requirements of the former, an approach based on Life Events is proposed. Within the context of this proposal, Life Events (here after LEs) are those situations that drive the citizen to interact with the administration in order to fulfill an obligation or execute a right. Thus, we can consider as "Life Events" situations such as getting certifications, paying a fine, getting married, moving, . . .

The first time the concept of LE about eGovernment was used is related to the eGovernment project. In that context, Life Events were defined as "situations of human beings that trigger public services". That definition is the starting point for our semantic definition of LE. This idea is reused in different official pages such as the Ontario's Official Site, Nova Scotia's one, the Irish eGovernment initiative and others. Those pages make use of the concept of LE to index and locate services according the citizen requests.

We make a step forward towards the definition of LE. By mean of semantic definitions and properties, an entire system is proposed to catalogue, search, discover, and orchestrate services in the domain. In the definition of LEs, documents play a relevant role. In any democratic administration, documents are the only prove that an operation has to been done and must be supported.

Taking into account the former considerations, we establish a semantic based definition for LE. These elements are going to play a main role in our case and they are expressed using semantic terms shared by the whole system. The definition of a LE includes the following items.

1. Task. Title for the considered operation. Folksonomies can play an interesting role as they provide support for semi-automatic enhancements of discovering services.
2. Description. High level description of the desired operation expressed in natural terms from the point of view of the citizen.
3. Input Documents. As previously stated, all operations carried out by the administration require some input document. Citizen is requested to provide a signed form in order to invoke the operation. This element plays a role similar to *preconditions* in some environments. In the considered case, we can identify as inputs documents, the current certification
4. Output Document. Of course, as a result of any performed operation, the PA in charge must provide an output expressed in terms of the ontology. This information will be put together into one or several documents. This output will vary its content from the expected document (i.e., a certification, a license, . . .) to information about the failure to get the expected document.
5. Scope. We must identify the scope of the operation (local, national, international, . . .) where we want the operation to be recognized.
6. Security Conditions. This is intended to express the conditions for the security mechanism involved during the whole process. This includes the identification of both parties, citizen and PAs, and also the way is stored by any agent involved that could be able to use it.
7. Cost. This will express the amount you have to pay for the requested operation and/or also the time it will take for the completion of the operation.
8. Version. Life Events can be modified and changes from one version to another one must be tracked.

These elements will be defined using the power of semantic expressions that will allow us to provide advanced services for discovering and orchestrating them. LifeEvents can also be tagged using well-know metadata standard already proposed and endorsed by relevant organizations such as from the CEN.

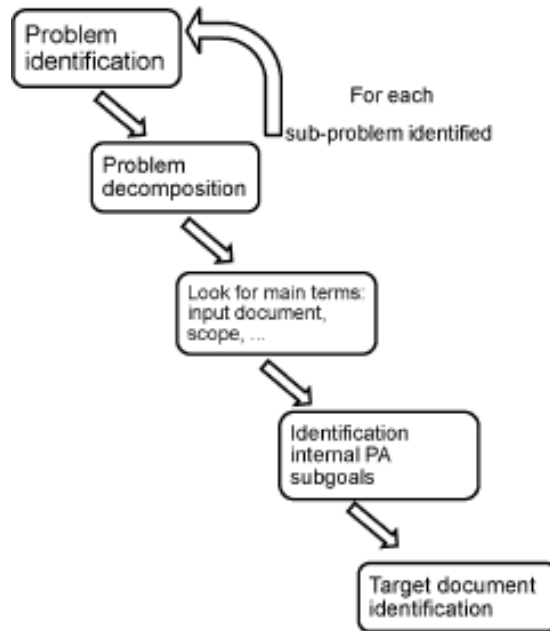


Figura 4.6: Schema for the definition of LE

So, we propose the transformation of final services as they are requested into new LEs expressed in terms of the semantic definition using the former items presented. Thus, the goals presented in the previous section about Tecut can be achieved. This schema is suitable for eGovernment field, or at least more suitable than in other environments, due to several reasons: all operations require some input document, the most common output in the service is a new document, there is no need (opportunity) for bargaining about services, there are limits and conditions very explicit about the data managing in terms of trustability and security (non-repudiation, privacy, integrity and confidentiality) and operations does not have real time constrains.

In order to transform common services into LE, expressed in the proposed terms, we must follow a simple methodology. For the sake of clarity, we are going to show the former by means of an example: the situation in which a citizen has to move to a new residence. This operation may require the collaboration of several different PAs and several processes the citizen does not have to be aware of. Thus, we propose the following schema (see Fig. 4.6).

1. Identify the problem and dealing features as PAs involved. Applied to our practical case, the task we are dealing with is the change of address for a citizen. The involved PAs are the cities council, of course, they should involve several offices or divisions but that should be transparent for the citizen.

2. Decompose the problem into different problems that may be solved in a single step, i.e., each step must produce as output a document meaningful for the citizen.
3. The considered operation in the example may involve one single operation and no subprocesses are relevant to the citizen.
4. For each identified subprocess, look for the input documents, scope and cost. These ones must be expressed in terms of the LE ontology. The input document required in our case is the certification of the current citizen address, the document to prove the new address and the signed request for the change. The scope for the operation is national. No cost is put on the citizen and no limitations are related to it.
5. Identify internal partial aims for citizens and PAs. These steps usually involve internal documents. They can be meaningless for the citizen but relevant for the administration. In our example, several steps can be identified: check for the correctness about the former address data, look for pending payments, update internal data, notify related PAs, and, finally, generate the certification for the new address.
6. Identify possible documents as possible final steps of the operation. In our case, the target document is the certification for the new address. Nevertheless, if problems arise, mainly related to some internal step, documents to notify those errors may be generated. Those documents will inform about problems due to pending payments, problems with legal constrains, . . . These documents must be included in the ontology.
7. Update all services and agents that may be aware of the new service.

Once all this information has been gathered and codified properly, it can be presented to the end user.

As a result we can identify in Tecut a classification of LE that enhances the accessibility from the point of view of the citizens. Navigating from the home page of the Portal, users can easily access a list of LE classified according to a taxonomy to choose the one best fits in their interests.

4.4.6 Semantic Life Events

Semantic plays a relevant role in this solution. By means of ontologies we are addressing a higher level of abstraction than the one based on raw data. To undertake the provision of an ontology we may use different languages. OWL (Ontology Web Language) a W3C Recommendation is the chosen one for our proposal. By using OWL, we are addressing a standard, solid and interoperable platform for the provision of this solution. Proposed approach takes advantage of the power of OWL to express the information relevant for the system. Nevertheless, we must keep in mind that OWL is just a tool to express knowledge

with all its potential and limitations. Thus, following Methontology, a FIPA recommended process to develop ontologies, one has been developed.

In this ontology, we have reused former already defined data representation. For example, for the definition of the citizen, one main class in the system, FOAF has been reused, and, to mark documents in the system, metadata in has been taken into consideration. This is part of a general philosophy leading toward the maximum possible agreement and reusability both of ontologies and software derived from the former.

On the other hand some limitations on the possibilities of OWL to express knowledge have been faced. In particular, OWL does not support relations that involve properties whose range is a class itself, just an individual from a particular class. This leads us into shortcomings in the definition of some relations (for example, we would like to establish a relation between an individual from the class LE and a subclass of “document”, not an individual from that class). This situation was overcome using a higher level of abstraction implicit in a single individual (the use of individual document belonging to the class document as a generic one with no information by itself).

Additionally and for the sake of consistency of current and future information in the system, some rules has been defined (see Fig 3): all LEs generate some Document (Rule 1), all LEs are supported by some PA (Rule 2), all Documents are issued by some PA, . . . Of course, lower level details about the conformance to local or national laws regarding document and legal procedures are not considered at this point and further implementations of the system should take care of it.

Once the ontology that describes the system is provided, the development of support for the access to these LEs must be faced. As the only possible interface is the Web page, all the logic and semantic processing is put on the Web Server. Nevertheless, the chosen approach is based on wrapping LEs with Semantic Web Services to define and to support them. The reasons for this decision are due to the wishes to provide a standardization of these definitions and the use of already developed software packages to deal with the information. The current state of art regarding this topic in the present moment it is quite unstable. Thus, we can find technologies designed to introduce semantic in Web Services that are emerging and others that may be in process of obsolescence. To meet our requirements, we decided to make use of WSDL-S. Main reasons to choose this option among other available possibilities are due to its simplicity but semantic power to express all required information. Other options were dismissed because of different reasons. OWL-S was seriously considered but it introduced a lot of overhead and it did not provide any clear advantage on WSDL-S, a much lighter technology. WSMO was also considered but the use of mediators does not really fit in the aim of this project.

Each LE drives the generation of a WSDL-S file describing it. *Inputs* and *outputs* in each *operation* included in the system, are defined in terms of the ontology developed. As the preconditions and effects are implicitly provided, respectively, by the inputs and the outputs, it is enough if the latter are stated. Thus, in our example, the LE “*moving*” is defined using a WSDL-S file. This one

declares, as inputs, documents identified previously. Accordantly, the output of the operation is defined also in terms of the same ontology and, in this case, involves the already indicated documents. Thus, it is quite simple to make compositions using a semantic reasoner as it only will have to link outputs and inputs expressed in the same terms from the same ontology.

Rule	Definition
Rule 1 $R_1 = \{\forall LE \exists Doc,$ $generates(LE) = Doc\}$	<pre> <owl:Class rdf:about="#LifeEvent"> <rdfs:subClassOf> <owl:Restriction> <owl:onProperty> <owl:FunctionalProperty rdf:ID="generates"/> </owl:onProperty> <owl:someValuesFrom> <owl:Class rdf:about="#Document"/> </owl:someValuesFrom> </owl:Restriction> </pre>
Rule 2 $R_2 = \{\forall LE \exists PA,$ $isSupportedBy(LE) = PA\}$	<pre> <owl:Class rdf:about="#LifeEvent"> <rdfs:subClassOf> <owl:Restriction> <owl:someValuesFrom rdf:resource="#PA"/> <owl:onProperty> <owl:InverseFunctionalProperty rdf:ID="isSupportedBy"/> </owl:onProperty> </owl:Restriction> </rdfs:subClassOf> </pre>

Figura 4.7: Rules defined in the system

Anyhow, we must keep the perspective that WSDL-S is just another tool to introduce semantic on LE and many others can be used. As a matter of fact, if required, it is possible to extend the WSDL-S to other technologies with little effort. In fact, some transformations can be done easily.

As a result of these design decisions, advanced ways for the composition and the discovery of services are possible within the project Tecut.

4.4.7 Conclusions and Future works

In general, eGovernment Web portals are evolving towards a semantic distributed and cooperative approach. In particular, the Marche Region presents a federate community where the discovery of services play a main role. This federated reality allows the sharing of digital services. The fair distribution of the latter saves time and costs. Regarding to the discovery processes, we propose a richer semantic description of services, this proposal considers also metadata to introduce a flexible and extensible LE representation.

Therefore, this paper presents an in deep review of how semantic can be applied to the provision services in the domain of eGovernment. Thus, using a Web portal to provide support for citizen needs, a LE-based approach has been provided.

Currently, two working lines within the frame of this project are under design. The first one is related to the enhancement of semi-automatic discovery mechanisms. In this way, a wiki-like tool is planned to support the construction of a folksonomy to tag services with human friendly information. On the other hand, a digital repository of LEs defined by external PAs is to be designed. In order to obtain full potential from semantic Web Services, also it is planned to provide mechanism to allow that other PAs may upload their own definition of LEs. This would turn out Tecut into a common repository of services from a widespread group of PAs. All LEs in the pool would also be available for citizen through Tecut.

The transformation of common services into LE-based ones has been proofed to be a not too complex process that clearly increases the functionalities and capabilities of the entire system. To unleash all possible functionalities, from the presented work, semantics are called to play a main role in the process of describing and accessing information and services.

4.4.8 References

1. Bekkers, V.: The governance of back office integration in e-government: Some dutch experiences. In Wimmer, M., Traumm-onlund, °
2. uller, R., Gr-A., Andersen, K.V., eds.: EGOV. Volume 3591 of Lecture Notes in Computer Science., Springer (2005) 12–25
3. : Tecut. Available at (2007) <http://www.tecut.it>.
4. UN: Global e-Government readiness report 2004. Towards access for opportunity. Web available (2007) <http://unpan1.un.org/intradoc/groups/public/documents/UN/UNPAN019207.pdf>.
5. Tambouris, E.: An integrated platform for realising one-stop government: The eGOV project. In: DEXA Workshop. (2001) 359–363
6. Gant, J., Gant, D.: Web portal functionality and state government e-service. In: System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on. (2002) 1627– 1636
7. UK GovTalk: e-GIF. Disponible en la Web (2004) <http://www.govtalk.gov.uk/>.
8. : Ontogov. Disponible en la Web (2005) <http://www.ontogov.com/>.
9. : Terregov. Disponible en la Web (2005) http://www.terregov.eupm.net/my_spip/index.php.
10. : The SemanticGov Project. Web available (2006) <http://www.semanticgov.org>.
11. : SW-Suomi. Web available (2007) <http://www.museosuomi.fi/suomifi>.

12. Wimmer, M.: Implementing a knowledge portal for e-government based on semantic modelling: the e-government intelligent portal (eip.at). In: Proceedings of the 39th Hawaii International Conference on System Sciences. (2006)
13. Corradini, F., Angelis, F.D., Ercoli, C., Polzonetti, A., Re, B.: Consideration to improve e-government infrastructure. In: Proceedings of the International Conference on e-Society. (2006)
14. Corradini et al.: Shared services center for e-government policy. In: e-GovInterop05. (2005) 140–151
15. Clercq, J.D.: Single sign-on architectures. In: Proceedings of the International Conference on Infrastructure Security. (2002) 40 – 58
16. Regione Marche: Piano d'azione regionale per la missione della carta raffaello. accordo programma quadro Societ dell'Informazione, delibere CIPE 36/02 e 17/03. (2003)
17. : egov: Online one-stop government. Web available (2005) <http://www.egov-project.org/>.
18. CEN: Guidance on the use of metadata in e-government. Web available (2004) <http://www.cenorm.be/cenorm/businessdomains/businessdomains/iss/cwa/cwa14859.asp>.
19. CEN: Dublin Core eGovernment Application Profiles. Web available (2004) <http://www.cenorm.be/cenorm/businessdomains/businessdomains/iss/cwa/cwa14860.asp>.
20. CEN: Guidance information for the use of dublin core in europe. Web available (2004) <ftp://cenftp1.cenorm.be/PUBLIC/CWAs/e-Europe/MMI-DC/cwa13988-00-2003-Apr.pdf>.
21. Gruber, T.: A translation approach to portable ontology specifications. Knowledge Acquisition (1993) 199–220
22. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering. Springer (2003)
23. W3C: Web ontology language. Web available (2004) <http://www.w3.org/2004/OWL/>.
24. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: From ontological art towards ontological engineering. Symposium on Ontological Art Towards Ontological Engineering of AAAI. (1997) 33–40
25. : the foaf project. Web available (2005) <http://www.foaf-project.org/>.
26. World Wide Web Consortium: Web service semantics -wsdl-s. Web available (2007) <http://www.w3.org/Submission/WSDL-S/>.

27. OWL-S Coalition: OWL-S: Semantic Markup for Web Services. Web available (2005) <http://www.daml.org/services/owl-s/1.1/>.
28. SDK WSMO working group: Wsmo. Web available (2005) <http://www.wsmo.org/TR/d2/v1.1/>.
29. : D30v0.1 Aligning WSMO and WSDL-S. Web available (2007) <http://www.wsmo.org/TR/d30/v0.1/>.

4.5 Un'esperienza di modellazione dei servizi di e-Government

Basato su materiale contribuito da:
Annalisa Barone e Paolo Di Pietro (Diviana e-consulting)

Arianna è l'approccio utilizzato da Diviana, società di e-consulting, per definire un modello standard atto a descrivere la semantica dei servizi di e-government. Tale modello è nato per condividere informazioni tra le Pubbliche Amministrazioni Locali Italiane.

Per descrivere semanticamente i servizi, il nostro approccio modella le ontologie usando lo Unified Modeling Language (UML[1]). Il modello è automaticamente convertito in un sito web SVG[3] semanticamente navigabile ed in una serie di Schemi di Definizione XML[2] (XSD) che descrivono la struttura dei dati utilizzati nei servizi, rappresentano il modello di comunicazione tra le diverse Amministrazioni e rendono possibile l'interoperabilità applicativa.

Un generatore automatico di front-end guidato dall'ontologia e completamente personalizzabile dall'Ente permette la standardizzazione delle relazioni fra i cittadini e la P.A. e la generazione di applicazioni XForms, che implementano interfacce utenti.

4.5.1 Introduzione

La diffusione di Internet ha portato ad un elevato livello di standardizzazione, rendendo possibile navigare tra milioni di siti che utilizzano tecnologie eterogenee. Sviluppatesi a partire dalla seconda metà degli anni Novanta, l'Interoperabilità applicativa, ossia l'insieme di tutto ciò di cui si ha bisogno affinché due o più applicazioni possano interagire tra loro per raggiungere uno specifico obiettivo, è oggi pervasiva in molte delle attività sulla Rete.

L'ulteriore passo in avanti che si vuole compiere è trasformare i siti da isole in arcipelaghi, in modo tale che abbiano leggi, regole e linguaggi comuni.

La P.A. Italiana è un arcipelago di Entità Amministrative ciascuna con la propria autonomia e con la necessità di interagire ed integrarsi con le altre. Ad esse si aggiungono il Governo Centrale e le Agenzie di Governo collegate.

In questo quadro si rivela cruciale la definizione di uno standard di comunicazione tra le diverse Amministrazioni. Tale standard non può essere imposto per legge, poiché ciò implicherebbe una riduzione dell'autonomia decisionale degli Enti garantita dalla Costituzione, ma deve essere realizzato attraverso un processo di concertazione tra i diversi attori, al fine di condividere la conoscenza.

Mettere a punto un modello condiviso per la P.A. è un processo estremamente complesso, che va gestito attraverso un approccio formale.

Arianna può essere raggiunta all'indirizzo <http://Arianna.diviana.net>.

4.5.2 L'Ontologia della Pubblica Amministrazione Locale

Il primo passo per gestire la complessità consiste nel condividere la conoscenza: il concetto che guida il nostro approccio in questa direzione è l'Ontologia della P.A.

L'Ontologia è la base per gestire le interazioni tra un gran numero di Entità Amministrative autonome, per identificare i concetti, le informazioni, i soggetti, i ruoli e le relazioni reciproche, dando a ciascuno sia una caratterizzazione semantica sia una descrizione della struttura intrinseca. Per raggiungere l'obiettivo della condivisione sono stati analizzati e classificati secondo diverse tassonomie i servizi offerti dalle PAL, descrivendo per ciascuno i fornitori e i clienti, in modo da individuarli e accedervi semplicemente.

L'Ontologia si fonda su uno specifico paradigma di soluzione, descritto utilizzando una base di conoscenza UML e caratterizzato da tre aspetti: la definizione di un modello di riferimento architetturale; la definizione di un pattern di interoperabilità; un approccio interamente orientato al riuso.

4.5.3 Architettura: un modello di riferimento

Allo scopo di definire un approccio semantico che possa ospitare diverse implementazioni specifiche, l'architettura che abbiamo definito è una generica suddivisione dei livelli logici.

Su questa architettura, l'utente richiede un servizio utilizzando un portale o un'applicazione di front-end. La richiesta passa da un Centro Servizi, che funge da intermediario e che, attraverso una serie di servizi infrastrutturali, è capace di identificare il destinatario finale cui indirizzarla e le modalità d'invio.

4.5.4 Pattern di Interoperabilità

E' stato così identificato un unico Pattern di interoperabilità in grado di descrivere tutti i servizi di e-government, nonché i differenti ruoli e le principali attività che si devono effettuare per permettere l'implementazione.

In sintesi, esso descrive il workflow delle attività che devono essere svolte dai vari ruoli, ogni attività è poi descritta attraverso workflow di più basso livello.

Il pattern può dunque essere applicato a qualunque servizio di e-government, perché è utilizzato in tutto il modello ed è referenziato da tutti i servizi.

4.5.5 L'approccio basato sul riuso

La stella polare del nostro progetto è sempre stato il concetto di riuso: di idee, di artefatti, di strutture organizzative, di processi. L'approccio scelto, infatti, permette di riutilizzare la conoscenza, aggregandola e rendendola fruibile ogni volta nelle forme e nei contenuti più opportuni per l'utente.

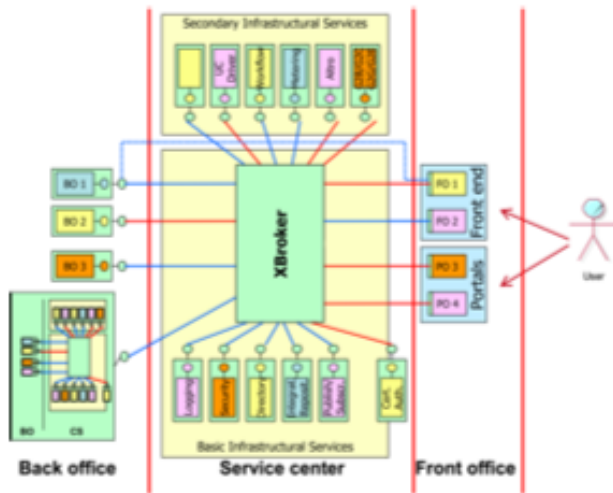
Lo scopo è quello di utilizzare un linguaggio comune, che abiliti la rappresentazione sia degli oggetti di business sia di quelli tecnici. Ma ancor di più c'è bisogno di un approccio metodologico, di un percorso completo che unisca tutta la conoscenza in un Repository comune, dove ciascun attore possa manifestare la propria e scoprire facilmente quella altrui.

4.5.6 La Base di Conoscenza UML

Il nostro modello può essere descritto anche nei termini di una Base di Conoscenza, contenente la descrizione di tutti processi di e-government da differenti punti di vista.

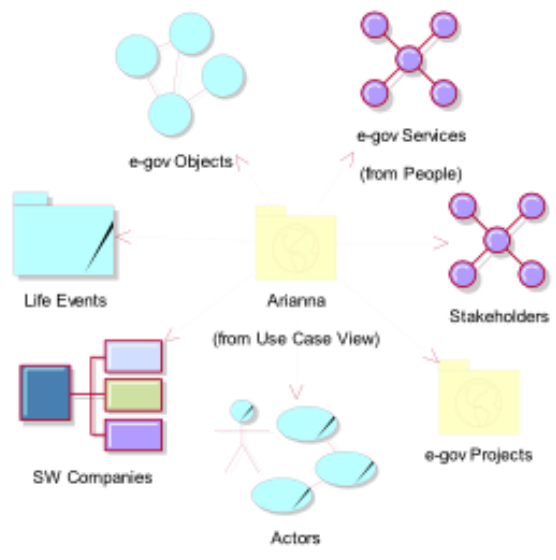
E' stato scelto UML come linguaggio per il Repository perché è uno standard di fatto nella progettazione di sistemi informativi e incorpora meccanismi di estendibilità.

La figura seguente mostra il menu principale del Repository.



A titolo di esempio, attraverso tale menu l'utente può accedere alla tassonomia dei servizi e-gov, navigando la quale si raggiunge il primo livello della tassonomia dei servizi.

Procedendo nell'area demografica, raggiunge il punto di vista della singola tassonomia, che raggruppa tutti gli artefatti collegati a quella particolare area, come mostrato in figura.

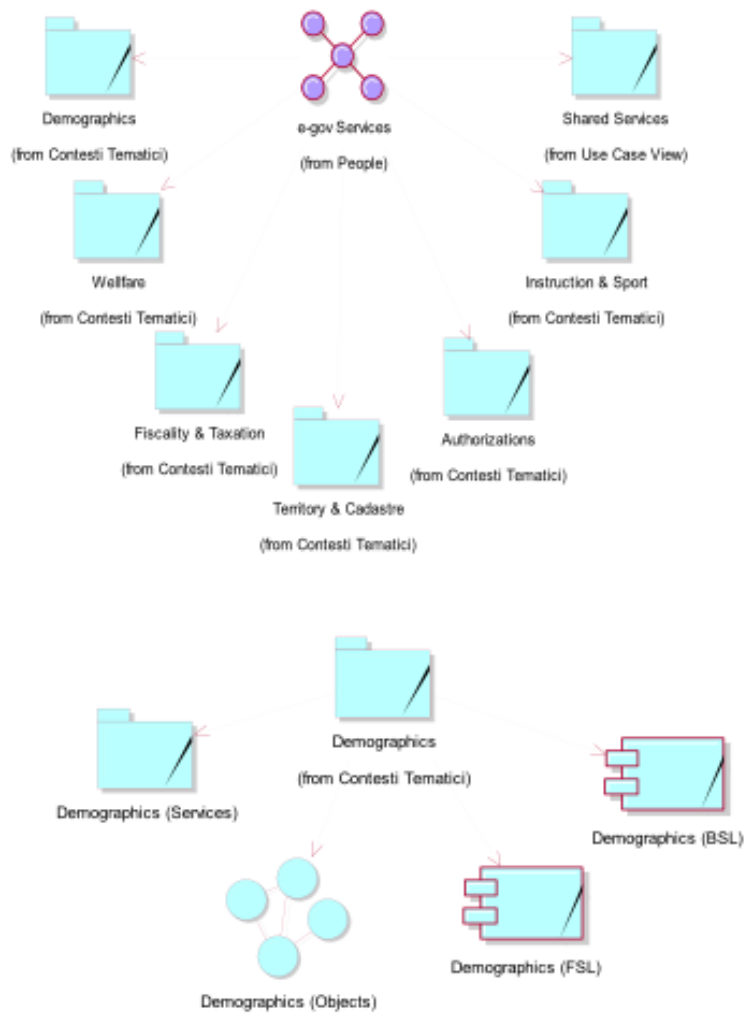


Quindi, selezionando Servizi, e dopo un ulteriore passo di selezione intermedio, raggiunge la lista che comprende tutti i servizi disponibili per la sottoarea selezionata.

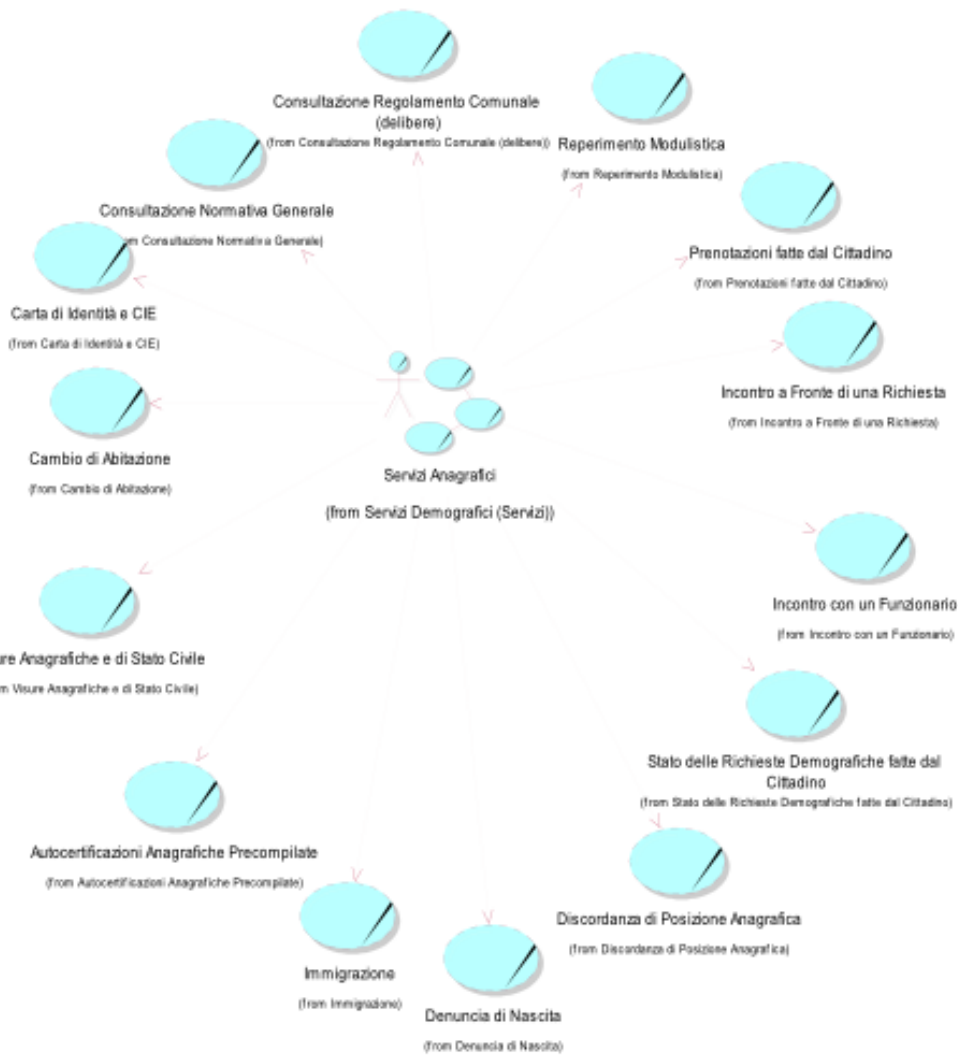
Selezionando un singolo servizio, raggiungiamo il suo Diagramma Principale, che mostra:

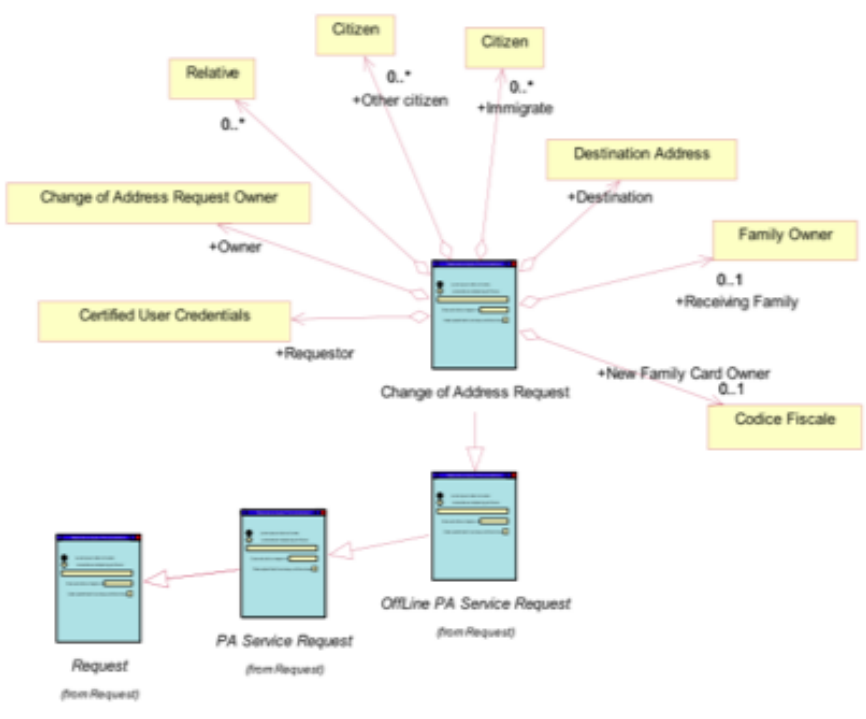
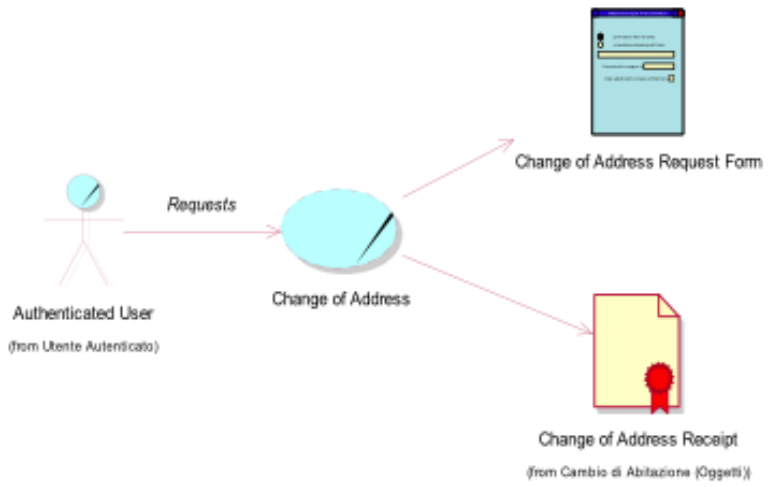
- I richiedenti ammessi, in questo caso un Utente Autenticato;
- Il Servizio stesso, rappresentato come Caso d'Uso;
- La Richiesta del servizio;
- La Ricevuta del servizio;
- Altri messaggi opzionali connessi al servizio;

Un esempio di questo diagramma è mostrato nella figura seguente.



A questo punto, cliccando una Richiesta o una Ricevuta si arriva direttamente al diagramma di classe UML che contiene la descrizione dettagliata dello specifico oggetto.





Naturalmente, ogni elemento contenuto nel diagramma è accompagnato da una dettagliata descrizione delle sue componenti. Il punto finale è un oggetto elementare che corrisponde ad un elemento semplice nel modello e che a sua volta mappa direttamente su uno specifico tipo semplice XSD.

La nostra assunzione è che la conoscenza sia reticolare e non gerarchica: il modello ha diverse vie di accesso, ciascuna delle quali evidenzia il punto di vista sulle informazioni contenute, di interesse per una specifica classe di utenti. Ogni elemento è modellato solo una volta nell'intera base di conoscenza, ma può essere raggiunto attraverso una combinazione di differenti percorsi liberamente selezionati dall'utente.

4.5.7 Pubblicazione della Base di Conoscenza

La pubblicazione della Base di Conoscenza è l'attività necessaria per rendere disponibile l'intero contenuto alle diverse classi di utenti (nuovi e abituali) e deve rendere disponibili i seguenti oggetti:

- *Una versione SVG*[3] del modello, che consenta di navigarlo logicamente sulla Rete;
- *Lo schema XML*, ossia la struttura informativa degli oggetti contenuti nel modello stesso;
- *La versione Xforms*, che realizza l'interfaccia utente dei servizi, completamente conforme al modello e direttamente personalizzabile da parte degli Enti a costi estremamente ridotti, attraverso il Generatore di front-end guidato dall'Ontologia.

Per completare il processo di pubblicazione sono stati generati dei Cataloghi di metadati, che contengono le definizioni dei servizi e degli oggetti. Questi sono il *Catalogo dei Servizi di e-government* ed il *Catalogo degli Oggetti di e-government*, applicazioni Internet che connettono tutti i singoli componenti e forniscono all'utente la possibilità di navigare tra le informazioni, passando facilmente tra i diversi punti di vista.

4.5.8 Benefici derivanti da questo approccio

La lista seguente descrive i punti salienti derivati dall'utilizzo di quest'approccio, i primi tre riportano considerazioni più politiche, mentre i rimanenti sono di tipo tecnico:

- La P.A. può personalizzare la componente di front-end senza la necessità di richiedere costosi interventi da parte delle aziende ICT;
- I cambiamenti normativi richiedono un unico intervento sul modello che può essere quindi redistribuito e riapplicato con costi minimi;
- Il time-to-market del cambiamento viene ridotto drasticamente e si avvicina allo zero teorico;

- L'intera applicazione di front-end deriva direttamente dal modello, assicurando così una corrispondenza al 100%;
- C'è una corrispondenza 1 a 1 tra il modello UML ed i prodotti derivati (XSD, Xforms, SVG, etc. etc.): ogni cambiamento al modello si riflette automaticamente nei prodotti;
- L'output del modello è una implementazione basata su web services di una richiesta di e-government, come descritta dal modello: l'XML risultante non è alterato dalla personalizzazione dell'interfaccia utente;
- Negli Xforms le informazioni complesse vengono collasate in modo da fornire all'utente una vista compatta espandibile quando necessario;
- Le varie alternative previste dal modello corrispondono a diverse possibilità offerte dalla interfaccia utente;
- L'ente può facilmente personalizzare l'interfaccia, arrivando ad eliminare completamente tutti gli elementi opzionali;

4.5.9 Alcuni numeri

Attualmente, il Repository contiene più di 250 servizi di e-government completamente modellati, raggruppati in 7 aree tematiche. Ci sono oltre 1.600 classi di oggetti[4], più di 19.000 relazioni tra oggetti, più di 1.780 diagrammi, per un totale di oltre 100.000 elementi di conoscenza.

In media, ogni servizio utilizza 124 diverse classi, mentre ciascuna classe è utilizzata mediamente 18 volte dai diversi servizi.

Riutilizzare una classe 18 volte è un risultato stupefacente: l'obiettivo iniziale di definire una interfaccia standard per i servizi ha portato all'effetto di ridurre i costi di sviluppo e di manutenzione, fornendo allo stesso tempo la possibilità di sviluppare software basato su componenti a tutte le aziende che dispongono delle necessarie competenze.

4.5.10 Prossimi passi

L'approccio descritto è stato utilizzato da diversi progetti di e-government nel periodo 2001-2006, ed è stato fatto proprio dal Progetto People.

Stiamo ora pianificando le possibili evoluzioni all'approccio proposto:

- Misurazioni e sistema metrico: rilevare alcune misure, che forniscano alle persone una miglior comprensione del contenuto della Base di Conoscenza e le supportino nel prendere decisioni.
- Gestione olistica dei progetti: vista la complessità dei progetti di e-government è necessario attivare un approccio olistico che coinvolga le persone nel processo decisionale.

- Tool per la verifica della completezza e della consistenza: aumentando la dimensione della Base di Conoscenza, è necessario poter disporre di strumenti che verifichino la completezza e la consistenza del suo contenuto.

L'interesse per i risultati fin qui ottenuti, sta anche portando allo sviluppo di diversi nuovi progetti che riusano ed estendono l'approccio e la Base di Conoscenza:

- Integrazione con un catalogo UDDI 3.0 - Repository di informazioni tecniche sulla istanziazione dei servizi
- Condivisione dell'approccio con Aitech - Assinform, associazione confindustriale italiana che raggruppa le aziende che sviluppano software per la P.A.

Bibliografia

- [1] UML: **U**nified **M**odeling **L**anguage. E' un linguaggio di modellazione sviluppato nella seconda metà degli anni 90 da un pool di aziende, coordinate dall'Object Management Group, uniformando le diverse esperienze e metodologie di rappresentazione Object Oriented in uso fino a quel momento.
- [2] XML: **eX**tensible **M**arkup **L**anguage descrive una classe[4] di oggetti chiamati *documenti XML* e descrive parzialmente il comportamento dei programmi software che li elaborano. Esso identifica inoltre una famiglia di linguaggi, descritti nel sito <http://www.w3.org>.
- [3] SVG: **S**calable **V**ector **G**raphics. Si tratta di una modalità utilizzata per descrivere oggetti grafici bidimensionali in maniera indipendente dallo strumento utilizzato per la visualizzazione. Ciò consente di mantenere il massimo livello qualitativo dell'immagine, senza sfocature o perdite di dettagli, consentendo all'utente di navigarne i contenuti e di zoommare a proprio piacimento. Per informazioni: <http://www.w3.org/Graphics/SVG>
- [4] Per gli inesperti di Object Oriented, il concetto di Classe può essere paragonato a quello di Entità.

Capitolo 5

Caso di studio: un'ontologia per il mondo del lavoro

5.1 Integrazione di modelli XSD per la PA che descrivono il mondo del lavoro

NOTA: questa sezione deve essere ancora elaborata e qui viene mostrato solo l'outline.

Basato su materiale contribuito da: Aldo Gangemi, Luca Torri, Amedeo Bogliaccino, Paolo Di Pietro, etc.

5.1.1 Introduzione

5.1.2 Requisiti di INPS e BCNL

5.1.3 Il modello dell'INPS per le “Comunicazioni obbligatorie”

5.1.4 Il modello della BCNL

5.1.5 Il lavoratore in Arianna

5.1.6 Il lavoratore nei modelli internazionali

5.1.7 Problemi

5.2 Proposta di metodologia per l'annotazione semantica di accordi di servizio

Basato su materiale contribuito da: Michele Trainotti e Marco Pistore, ITC Trento

5.2.1 Introduzione

Nel quadro dell'eGovernment l'interoperabilità a livello tecnico è stata indirizzata in modo efficace dalle specifiche SPCoop emanate dal Centro Nazionale per l'Informatica nella Pubblica Amministrazione (CNIPA). Per garantire pieno supporto all'interoperabilità è necessario affiancare all'interoperabilità a livello tecnico anche l'interoperabilità a livello semantico, come peraltro è già previsto dalle specifiche tecniche sopra citate. La definizione di layer semantico è attualmente oggetto di approfondimento sia a livello del CNIPA che nel contesto del progetto ICAR.

Nella sperimentazione oggetto di questo lavoro si vuole procedere alla definizione di una metodologia per l'annotazione semantica di accordi di servizio elaborata sulla base di un specifico scenario applicativo di riferimento nel contesto di ICAR. In particolare lo scenario di sperimentazione applicativa di riferimento è costituito dal task AP4 che prevede la fruizione in cooperazione applicativa dei servizi offerti dal dominio di cooperazione della Borsa Continua Nazionale del Lavoro (BCNL). Lo stesso dominio di cooperazione è anche utilizzato come caso di studio al tavolo di lavoro CNIPA su "Semantica e Metadati".

Il dominio applicativo della BCNL definisce un insieme di servizi a supporto della mobilità lavorativa esportati in cooperazione applicativa dal Ministero del Lavoro. In sintesi l'insieme dei servizi sono tesi a fornire degli strumenti per definire l'incontro tra domanda e offerta lavorativa.

Il documento è strutturato nelle sezioni:

- "Accordo di Servizio": definisce il ruolo dell'accordo di servizio CNIPA/ICAR nel contesto della interoperabilità e cooperazione applicativa
- "Semantica": illustra la proposta di metodologia di definizione del layer semantico nel contesto dell'interoperabilità e cooperazione applicativa
- "Strumenti": descrive gli strumenti relativi alla formalizzazione del layer semantico individuati nella sperimentazione della metodologia illustrata
- "Un esempio pratico": propone la descrizione dell'applicazione della metodologia illustrata su un caso concreto
- "Appendice A": nella sezione viene riportato un estratto dell'accordo di servizio definito durante la fase di sperimentazione

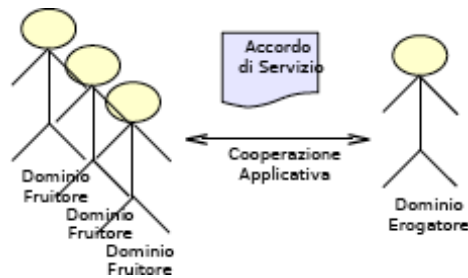


Figura 5.1: Ruolo dell'accordo di servizio

5.2.2 Accordo di Servizio

Le specifiche SPCoop definiscono come elementi fondanti la cooperazione applicativa il *registro dei servizi*, la *porta di dominio*, la *busta di eGovernment* e l'*accordo di servizio*. Il *registro dei servizi* è l'elemento infrastrutturale responsabile della localizzazione dei servizi erogati in cooperazione operativa. La *porta di dominio* è l'elemento infrastrutturale responsabile della mediazione tra rete SPCoop ed il sistema informativo interno. La *busta di eGovernment* rappresenta il protocollo di comunicazione utilizzato in cooperazione applicativa. Infine è previsto che per fruire dei servizi erogati in cooperazione applicativa i soggetti cooperanti debbano definire un *accordo di servizio* che disciplina le modalità di erogazione e fruizione del servizio.

Il ruolo dell'*accordo di servizio* è fondamentale nel contesto della cooperazione applicativa, esso infatti definisce il contratto di collaborazione tra i due o più attori cooperanti. In Figura 1 è illustrato come l'accordo di servizio serva da "contratto" per disciplinare la fruizione da parte dei dominio fruitore dei servizi erogati in cooperazione applicativa dal dominio erogatore. L'*accordo di servizio* è strutturato in una parte comune e tante parti specifiche quanti sono i fruitori del servizio. La metodologia di definizione del layer semantico su cui si concentra questo documento riguarda principalmente la parte comune dell'accordo di servizio, di cui riportiamo quindi la struttura.

La parte comune dell'accordo di servizio prevede: la definizione delle specifiche di interfaccia, la definizione delle specifiche delle conversazioni, la definizione degli schemi e delle ontologie di riferimento ed infine la definizione delle informazioni di egov. In Figura 2 è rappresentata la parte comune dell'accordo di servizio così come definito dalle specifiche CNIPA, si rimanda a "*Sistema pubblico di cooperazione: ACCORDO DI SERVIZIO*" per verificare la definizione completa dell'accordo di servizio CNIPA.

L'accordo di servizio ICAR consiste in un insieme di file, distribuiti separatamente o raccolti in un archivio compresso secondo la codifica zip, aderenti a vari formati standard di definizione di specifiche caratteristiche dell'accordo, come WSDL, ed altri formalismi realizzati adhoc, come WSBL. Tra i vari file parte dell'accordo, uno ha la funzione di manifest. Se l'accordo è distribuito come archivio, il manifest è incluso nell'archivio con il path META-INF/manifest.xml.

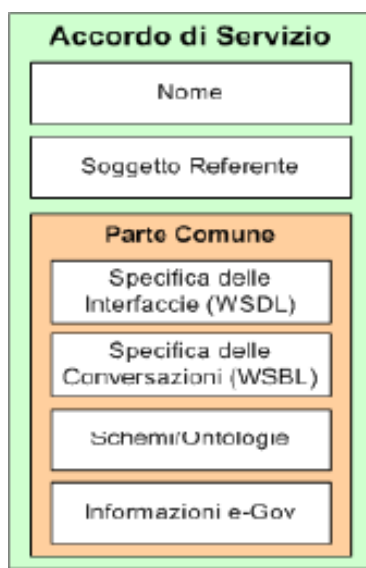


Figura 5.2: Schema della parte comune dell'accordo di servizio ICAR

Si rimanda a “INF1: Specifiche Accordo di Servizio” per verificare la definizione completa dell'accordo di servizio ICAR.

La definizione dell'accordo di servizio si è concretizza nell'analisi della documentazione dei servizi erogati in cooperazione applicativa che avevamo a disposizione e nella sintesi e scrittura dell'accordo di servizio completo nelle sue varie parti. Nella sezione “Appendice A” viene riportato un stralcio dell'accordo di servizio definito per il dominio della BCNL.

5.2.3 Semantica

Nell'ambito dell'interoperabilità e della cooperazione applicativa la semantica è lo strumento che garantisce la possibilità che sistemi informativi distribuiti, basati su differenti concetti associati alle informazioni scambiate, possano collaborare in modo automatico anche se sono stati progettati in modo indipendente. L'obiettivo è di fare in modo che le fonti informative siano in rete e che l'informazione scambiata sia automaticamente comprensibile e utilizzabile da applicazioni non coinvolte nella loro creazione. In sintesi serve per dare un significato preciso al flusso informativo scambiato tra due sistemi consentendo di risolvere eventuali ambiguità interpretative e per individuare corrispondenze e differenze fra concetti in domini diversi.

Una modalità per definire il *layer semantico* nel contesto della cooperazione applicativa è quella di introdurre delle annotazioni semantiche nell'*accordo di servizio*. La definizione del *layer semantico* prevede che a ogni singolo dominio amministrativo sia responsabile di definire una propria ontologia quanto meno

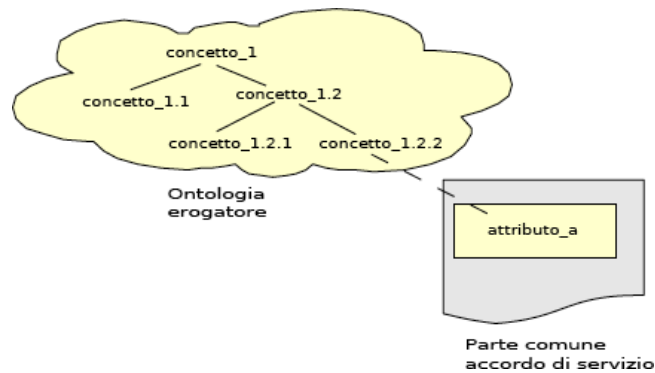


Figura 5.3: Annotazioni semantiche nell'accordo di servizio

delle informazioni e dei servizi erogati in cooperazione applicativa. A livello metodologico si propone di arricchire l'*accordo di servizio* con gli opportuni riferimenti semantici ai concetti espressi nell'ontologia di riferimento tramite delle annotazioni insistenti sulla parte comune dello stesso. In Figura 5.3 è riportato un esempio che illustra a livello concettuale l'approccio di annotazione semantica dell'accordo di servizio. In particolare nella figura sono evidenziati l'ontologia del dominio erogatore e l'accordo di servizio. Nella figura è evidenziata l'annotazione semantica di attributo per il quale nella parte comune dell'*accordo di servizio* viene definita la collocazione rispetto all'ontologia definita.

Una seconda modalità non necessariamente alternativa all'annotazione dell'accordo di servizio permette di prendere in considerazione le ontologie sia del dominio erogatore che del dominio fruitore e di andare a individuare le funzioni che consento di mediare tra i concetti definiti nelle due ontologie citate. In Figura 5.4 è riportato un esempio che illustra a livello concettuale l'approccio di annotazione definizione delle funzioni di mediazione. In particolare nella figura sono evidenziati l'ontologia del dominio erogatore, l'ontologia del dominio fruitore e l'accordo di servizio. Nella figura è evidenziata l'annotazione semantica di attributo per il quale nella parte comune dell'*accordo di servizio* viene definita la collocazione rispetto all'ontologia del dominio erogatore mentre nella parte specifica viene definita la collocazione rispetto all'ontologia del dominio fruitore. Le funzione di mediazione è identificata dalla freccia che unisce i concetti tra le due ontologie.

In questo documentoci si concentra sull'approccio che prevede l'annotazione dell'accordo di servizio.

Il processo di sperimentazione è stato diviso tre macro-fasi:

1. definizione dell'accordo di servizio
2. definizione dell'ontologia di dominio
3. annotazione dell'accordo di servizio rispetto all'ontologia di dominio erogatore

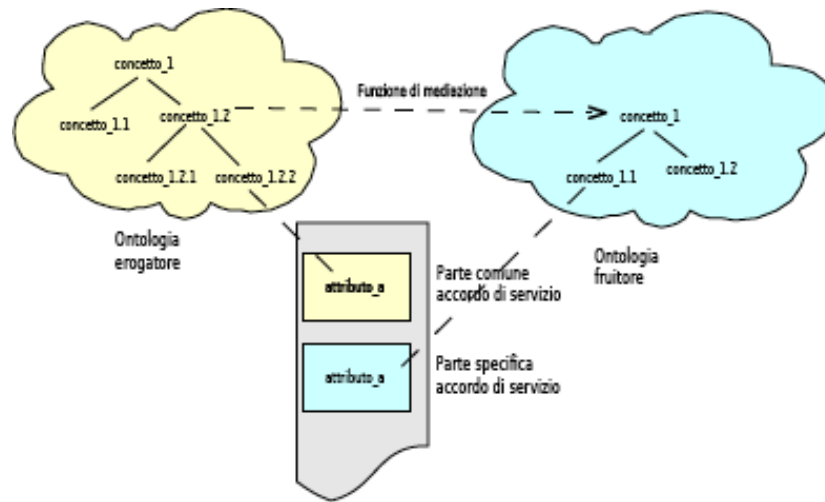


Figura 5.4: Mediazione fra ontologie

La definizione dell'accordo di servizio rappresenta di fatto la definizione del dominio di cooperazione applicativa. Questa fase ci ha permesso di individuare il modello dati, gli oggetti, le operazioni e l'insieme di servizi costituiscono insieme il dominio di cooperazione applicativa.

La definizione dell'ontologia di riferimento permette di catalogare e strutturare da un punto di vista semantico l'insieme dei concetti che sono utilizzati nel contesto del dominio di cooperazione applicativa.

L'annotazione degli elementi contenuti nell'accordo di servizio consente di legare elementi concreti con concetti semantici. Di fatto è lo strumento che permette di definire l'interpretazione autentica del dominio di cooperazione applicativa.

La definizione dell'accordo di servizio e la definizione dell'ontologia sono in generale del tutto indipendenti. E' importante sottolineare che cercare di definire l'ontologia senza aver chiari i confini di quello che si vuole concettualizzare pu risultare tal volta nella definizione di un ontologia di troppo di alto livello e tal altra nella definizione di un ontologia di troppo basso livello. Nella sperimentazione si è proceduto nella definizione dell'ontologia di riferimento partendo dall'accordo di servizio per garantire di avere chiaro il quadro di riferimento su cui si voleva operare. Questo ci ha permesso di utilizzare un approccio top down che partendo nella individuazione dei soli concetti semantici di alto livello che risultano essere parte del dominiodi cooperazioneapplicativa.

La sperimentazione ha avuto come oggetto l'annotazione semantica dell'insieme di servizi erogati in cooperazione applicativa della Borsa Continua Nazionale del Lavoro (BCNL). La definizione dell'ontologia di riferimento si è concretizzata in un processo a spirale che ha permesso di definire ed arricchire con passaggi successivi l'ontologia stessa dell'insieme degli elementi concettuali ri-

levanti dal punto di vista della cooperazione applicativa. Nella definizione del layer semantico dell'accordo di servizio gli aspetti da annotare sono: i dati, i servizi ed i processi. L'annotazione dei dati consente di dare la definizione formale del significato del dato e quindi consente di specificarne l'interpretazione autentica. L'annotazione dei servizi consente di generalizzare l'insieme degli stessi rispetto a opportune classi di servizio sul cui significato non ci sono ambiguità. Così' come per l'annotazione dei servizi, l'annotazione dei processi mira a definire generalizzare l'insieme degli stessi rispetto ad opportune classi di processo non ambigue. Inoltre è importante annotare eventuali ambiguità nell'interpretazione dei vari elementi dell'accordo di servizio attraverso la definizione un un insieme opportuno di vincoli formali. Infine è opportuno legare tutte le ontologie parziali sottese alle diverse fasi di annotazione in un'ontologia complessiva dell'accordo di servizio, chiamata ontologia di dominio.

Di seguito in questa sezione vengono dati dei chiarimenti rispetto alle diverse ontologie parziali risultanti dal processo di definizione del layer semantico dell'accordo di servizio.

Ontologia dei dati

L'ontologia dei dati identifica i concetti principali che caratterizzano la struttura dati espressa a livello del WSDL definitorio. Qui vengono raccolti i concetti principali di cui si parla nel dominio di cooperazione applicativa e le relazioni tra essi.

Ontologia dei vincoli

L'ontologia dei vincoli identifica i vari tipi di vincolo che possono sussistere tra i diversi elementi, un esempio è costituito dai vincoli che ci sono tra gli elementi di un messaggio inviato ad un servizio da un fruitore. Nell'ontologia è stato definito il concetto di vincolo, mentre tutti i vincoli identificati nel dominio di cooperazione vengono modellati come individui della classe vincolo.

Ontologia dei servizi

L'ontologia dei servizi identifica l'insieme delle classi di servizio che generalizzano i servizi erogati in cooperazione applicativa e esplicita la relazione tra classe di servizio e servizio. Per ogni classe di servizio vengono identificate e dettagliate le operazioni supportate, le stesse che vengono modellate come individui del concetto di *operazione* definito nel contesto ontologico di riferimento. I servizi esportati in cooperazione applicativa vengono definiti come individui appartenenti alle classe di servizio opportuna tra quelle individuate, in modo da evidenziare il legame di appartenenza tra servizio erogato e una classe di servizi specifica.

Ontologia dei processi

L'ontologia dei processi identifica l'insieme delle classi di processo che generalizzano i processi erogati in cooperazione applicativa e rende esplicita la relazione di appartenenza tra classe di processo e processo.

Ontologia di dominio

L'ontologia di dominio modella il dominio di cooperazione identificando i servizi ed i processi esportati in cooperazione applicativa. Essa è a tutti gli effetti l'ontologia che contiene le varie sottoontologie sin qui definite.

5.2.4 Strumenti

Come abbiamo visto, l'interoperabilità a livello semantico trova nella definizione di ontologie e di efficienti tecniche di matching lo strumento principale per la sua realizzazione. Allo stato attuale esistono diverse tecniche e linguaggi utilizzabili in ambito semantico. Dal punto di vista dei linguaggi i lavori più interessanti sono quelli relativi a OWLS e SAWSDL.

Dal punto di vista tecnico lo strumento di riferimento per l'annotazione semantica è costituito da SAWSDL, uno standard proposal condivisa sia dal mondo scientifico che da quello industriale.

OWL

Web Ontology Language (OWL) è un linguaggio per definire ontologie web basato sui precedenti OIL e DAML+OIL. Il riconoscimento di OWL come W3C recommendation ha portato allo sviluppo di un crescente insieme di tools e tecniche di ricerca che come obiettivo l'introduzione della semantica negli ambienti di produzione.

OWL è stato progettato e sviluppato con l'intento di unificare il modo di processare il contenuto semantico e l'informazione web; è basato su XML e questo ne facilita l'uso nell'ambito di sistemi eterogenei e distribuiti.

SAWSDL

SAWSDL o Sematic Annotation for WSDL and Schema definisce come arricchire di annotazioni semantiche varie parti di un documento WSDL seguendo le raccomandazioni definite nel WSDL 2.0/WSDL 1.1 extensibility framework. Le annotazioni semantiche sono definite attraverso l'utilizzo di attributi aggiuntivi rispetto alla grammatica standard di WSDL che permettono di legare gli elementi propri del linguaggio all'ontologia di riferimento, in questo molti dei concetti introdotti sono mutuati da WSDL.

SAWSDL si concentra sull'annotazione della parte astratta definita dal WSDL e non entra nella specifica dell'implementazione dello stesso, in particolare sono definiti tre attributi: *modelReference*, *liftingSchemaMapping* e *loweringSchemaMapping*. L'attributo *modelReference* permette definire l'associazione tra un

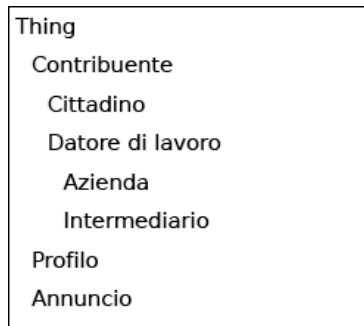


Figura 5.5: Ontologia dei dati

componente WSDL ed un concetto definito nell'ontologia di riferimento. Gli attributi *liftingSchemaMapping* e *loweringSchemaMapping* sono aggiunti alla parte di definizione del XML Schema e consentono di specificare la relazione tra il documento XML e l'ontologia di riferimento.

5.2.5 Un esempio pratico

In questa sezione vengono presentate le ontologie parziali di dati, servizi, processi e vincoli e dell'ontologia di dominio definite rispetto all'analisi dell'accordo di servizio relativo alla Banca Comune Nazionale del Lavoro di cui è definito uno stralcio nella sezione "Appendice A".

Ontologia dei dati

Nel contesto della caso di studio sono stati identificati concetti come *Profilo*, *Annuncio*, *Azienda*, *Cittadino*. In Figura 5.5 è proposto uno schema dell'alberatura ontologica risultante.

Oltre all'alberatura nell'ontologia è possibile definire aspetti di più basso livello quale l'insieme degli attributi associati ad un dato concetto. Con una rappresentazione di dettaglio è possibile presentare anche l'insieme degli attributi del concetto di *azienda* è riportato è riportato in Figura 5.6.

Ontologia dei vincoli

Nel contesto del caso di studio sono stati evidenziati un insieme insistenti sui messaggi scambiati tra il dominio erogatore ed il dominio fruitore. Ad esempio il è importante che il messaggio di cancellazione di un profilo faccia riferimento ad un vincolo presente nel dominio di cooperazione dell'erogatore del servizio. In Figura 5.7 è proposto uno schema dell'alberatura ontologica risultante.

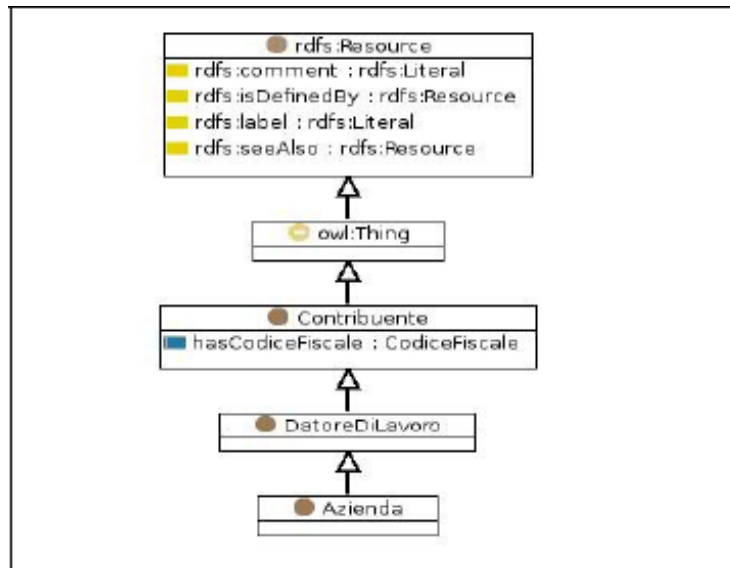


Figura 5.6: Rappresentazione di dettaglio del concetto Azienda



Figura 5.7: Ontologia dei vincoli

Ontologia dei servizi

Nel contesto del caso di studio l'insieme dei servizi offerti nel dominio di cooperazioni applicativa sono stati individuate le classi di servizio: il *Borsa*, il *Indice* ed il *Casella postale*. Per quanto concerne le operazioni supportate, per il concetto di *Borsa* sono state individuate le operazioni *Riponi*, *Cerca*, *Togli* e *Guarda*. Infine gli individui identificati sono: la *Borsa degli annunci* di lavoro, la *Borsa dei profili* dei cittadini, l'*indice dei cittadini*, l'*indice delle aziende*, l'*indice degli intermediari*, il servizio di *Casella postale per i cittadini*, il servizio di *Casella postale per le aziende* ed il servizio di *Casella postale per gli intermediari*. In Figura 5.8 è proposto uno schema dell'alberatura ontologica risultante.

Ontologia di dominio

In riferimento al caso di studio, si definisce il concetto di BCNL come quell'elemento concettuale che eroga i servizi ed i processi riportati nell'alberatura ontologica di Figura 5.9.

5.2.6 Appendice A

In Figura 5.10 è riportato il *manifest.xml* dell'accordo di servizio di esempio definito nel contesto della sperimentazione.

In Figura 5.11 è riportato uno stralcio del WSDL definitivo dello scenario di sperimentazione definito come specificato in “Sistema pubblico di cooperazione: Accordo di Servizio, Versione 1.0”, CNIPA, 14 Ottobre 2005”.

In Figura 5.12 è riportato il WSDL concettuale dello scenario di sperimentazione definito come specificato in “Sistema pubblico di cooperazione: Accordo di Servizio, Versione 1.0”, CNIPA, 14 Ottobre 2005”.

Thing

Operazione

- CercaOggetto
- RiponiOggetto
- GuardaOggetto
- TogliOggetto
- LeggiMessaggio
- CancellaMessaggio
- InviaMessaggio
- AggiungiRiferimento
- CercaRiferimento
- RimuoviRiferimento
- DettaglioRiferimento

Borsa

- has CercaOggetto
- has RiponiOggetto
- has GuardaOggetto
- has TogliOggetto
- Borsa Annunci
- Borsa Profili

Indice

- has AggiungiRiferimento
- has CercaRiferimento
- has RimuoviRiferimento
- has DettaglioRiferimento
- Indice Cittadino
- Indice Datore di Lavoro
- Indice Intermediario

Casella Postale

- has LeggiMessaggio
- has CancellaMessaggio
- has InviaMessaggio
- Casella Postale Cittadino
- Casella Postale Datore di Lavoro
- Casella Postale Intermediario

Figura 5.8: Ontologia dei servizi

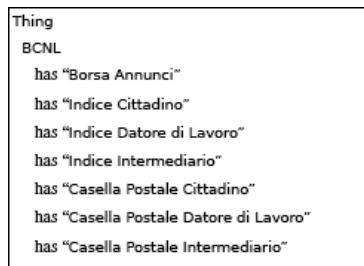


Figura 5.9: Ontologia di dominio

```

<?xml version="1.0" encoding="UTF-8"?>
<accordo-servizio nome="BCNL_ADS"
  descrizione="Accordo di servizio servizio Banca Continua Nazionale del Lavoro">
  <oggetto-referente tipo="SPC" nome="BCNL" />
  <specifica-interfaccia
    wsdl-definitori="BCNL_ADS/definitorio.wsdl"
    wsdl-concettuale="BCNL_ADS/concettuale.wsdl"/>
  <specifica-conversazioni
    wsbl-definitori="BCNL_ADS/definitorio.wsbl"
    wsbl-concettuale="BCNL_ADS/concettuale.wsbl"/>
  <catalogo-schemi-ontologie riferimento="BCNL_ADS/dominio.owl"/>
  <informazioni-egov riferimento="BCNL_ADS/informazioniEGov.xml" />
</accordo-servizio>

```

Figura 5.10: manifest.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.lego-lab.org/semantic/icar/lavoro/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name="interfacciaLogicaErogatore"
  xmlns:bcnlOnto="http://www.lego-lab.org/icar/lavoro/dominio.owl"
  targetNamespace="http://www.lego-lab.org/icar/lavoro/"
>
  <wsdl:types>
    <xsd:schema
      xmlns:tns="http://www.lego-lab.org/semantic/icar/lavoro/"
      targetNamespace="http://www.lego-lab.org/icar/lavoro/"
      attributeFormDefault="qualified">
      <xsd:import
        namespace="http://www.lego-lab.org/icar/lavoro/"
        schemaLocation="Lavoro_ADS/schemaLavoratore.xsd"/>
      [...]
      <xsd:complexType name="risposta_RichiestaRispostaSincrona_acknowledge_Type"
        sawSDL:modelReference="bcnlOnto#Acknowledge">
        <xsd:sequence>
          <xsd:element name="notifica" type="tns:notificaRisposta" />
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="errore_RichiestaRispostaSincrona_errore_Type"
        sawSDL:modelReference="bcnlOnto#ErroreApplicativo">
        <xsd:sequence>
          <xsd:element name="errore" type="tns:erroreRisposta" />
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="richiesta_RichiestaRispostaSincrona_insertProfilo_Type"
        sawSDL:modelReference="appsVincoli#vincolo_profilo_non_presente">
        <xsd:sequence>
          <xsd:element name="id" type="tns:CodiceFiscale" />
          <xsd:element name="lavoratore" type="tns:lavoratore" />
        </xsd:sequence>
      </xsd:complexType>
      [...]
    </xsd:schema>
  </wsdl:types>
  <!--
    definizione dei messaggi
  -->
  <!-- insertProfilo -->
  <wsdl:message name="richiesta_RichiestaRispostaSincrona_insertProfilo_Msg">
    <wsdl:part name="richiesta" type="tns:richiesta_RichiestaRispostaSincrona_insertProfilo_Type"/>
  </wsdl:message>
  <wsdl:message name="risposta_RichiestaRispostaSincrona_insertProfilo_Msg">
    <wsdl:part name="risposta" type="tns:risposta_RichiestaRispostaSincrona_acknowledge_Type"/>
  </wsdl:message>
  <wsdl:message name="errore_RichiestaRispostaSincrona_insertProfilo_Msg">
    <wsdl:part name="errore" type="tns:errore_RichiestaRispostaSincrona_errore_Type"/>
  </wsdl:message>
  [...]
</wsdl:definitions>

```

Figura 5.11: Stralcio di WSDL defintorio

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tnc="http://www.lego-lab.org/semantica/icar/lavoro/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name="interfacciaCocettualeErogatore"
  xmlns:bcnlOnto="http://www.lego-lab.org/icar/lavoro/ dominio.owl"
  targetNamespace="http://www.lego-lab.org/icar/lavoro/">

  <!--
    import wsdl definitorio
  -->
  <wsdl:import
    namespace="http://www.lego-lab.org/icar/lavoro/"
    location="Lavoro_ADS/definitorio.wsdl">
  </wsdl:import>

  <wsdl:portType name="IndiceRicerca_PT">
    <wsdl:operation name="insertProfilo"
      sawsdl:modelReference="bcnlOnto#/Borsa/BorsaProfili/ConferisciProfilo"/>
    <wsdl:input
      message="tnc:richiesta_RichiestaRispostaSincrona_insertProfilo_Msg" />
    <wsdl:output
      message="tnc:risposta_RichiestaRispostaSincrona_insertProfilo_Msg" />
    <wsdl:fault name="ErroreApplicativo"
      message="tnc:errore_RichiestaRispostaSincrona_insertProfilo_Msg"/>
    </wsdl:operation>

    <wsdl:operation name="removeProfilo"/>
    [...]
  </wsdl:operation>
</wsdl:portType>

  <wsdl:portType name="IndiceLocalizzazione_PT">
    [...]
  </wsdl:portType>

```

Figura 5.12: Stralcio di WSDL concettuale

5.3 Integrazione dei modelli usando l'ontologia di riferimento

Riprende le linee-guida e le applica all'integrazione dei modelli presentati, in riferimento all'architettura proposta dal metodo di annotazione semantica.

Capitolo 6

Gestione degli aspetti semantici dei servizi nella PA

Questo capitolo è ancora da definire

6.1 Servizi di gestione della semantica nella PA

- infrastruttura CNIPA, repository e catalogo dei servizi. *Documenti da integrare: ...*

6.2 Esperienze e linee-guida per la formazione da parte del CNIPA, della ricerca e delle aziende

- Contributi dai membri del GdL: CNIPA, Università, CNR, IBM, Oracle, altre aziende, ...