

Modello di composizione e transizione tra unità astratte di interazione

Survey

Nel presente documento si riassumono i contenuti del deliverable "D2.4_2 Modello di composizione e transizione tra unità astratte di integrazione" relativo all'attività di ricerca *RI 2.4 Definizione di Modelli per la caratterizzazione degli stadi di interazione tra utenti e servizi e delle modalità di transizione tra stati diversi*, nell'ambito del secondo Obiettivo Realizzativo (OR 2) "Studio di servizi e contenuti intelligenti".

Come mostrato in Figura 1, l'attività di ricerca è stata finalizzata alla definizione di un modello di descrizione ed annotazione della Logica di Business (LdB) di un servizio SAPI. Partendo dalla definizione delle caratteristiche di una activity, ovvero della componente di business fondamentale di un servizio intelligente, si procede alla descrizione dei meccanismi di composizione delle activity a formare un workflow di composizione di un e-service SAPI. Questa attività di modellazione getterà le basi per la definizione del modello di transizione tra le unità astratte di interazione (AIU). Infatti, pur rimanendo nell'ottica della separazione degli strati di composizione di un servizio intelligente, la transizione tra AIU è implicitamente scandita dall'avanzamento del flusso di esecuzione di una logica di business.

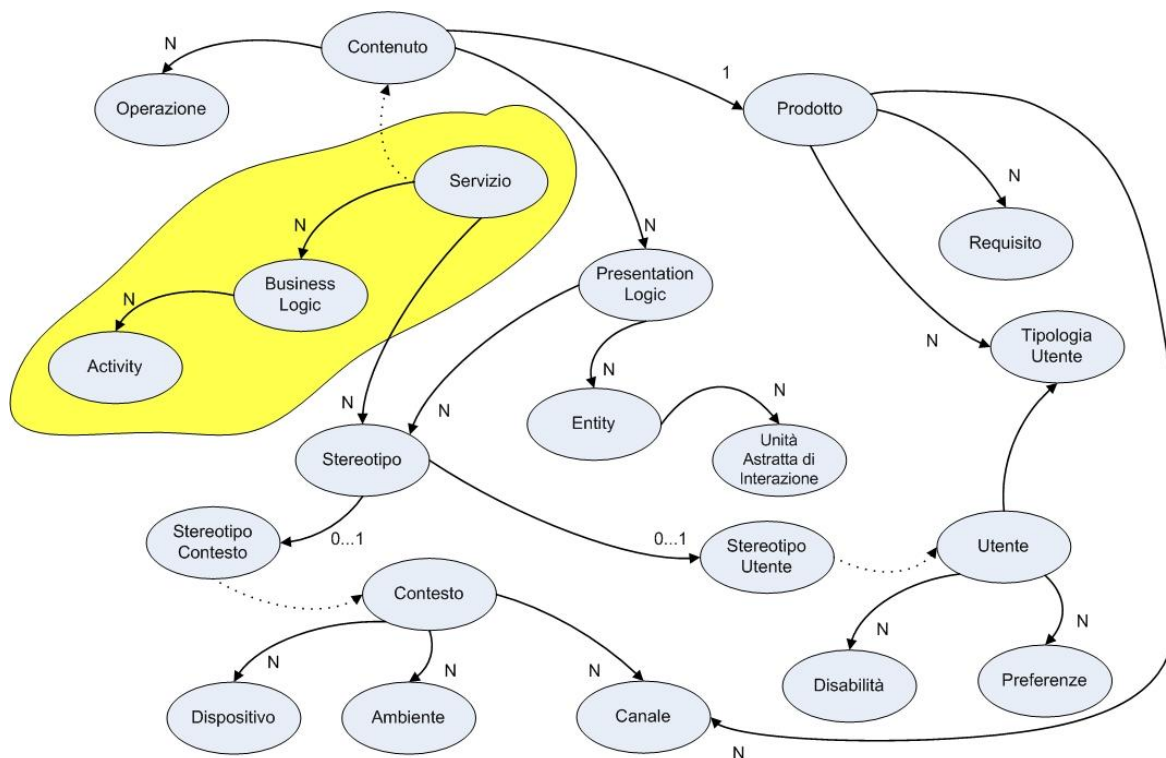


Figura 1: La Business Logic nel dominio SAPI.

La modellazione della logica di business è una parte essenziale dello sviluppo dei servizi intelligenti che saranno erogati dalla piattaforma SAPI. Essa consente di catturare le linee guida che regolano il processo di business, definendo i passi logici necessari ad una corretta interazione dell'utente col sistema. Ad un alto livello di astrazione la LdB di un servizio web può essere vista come un insieme di attività destinate a produrre, a partire da una serie di input, uno o più output destinati direttamente all'utente finale. La business logic regola, quindi, le operazioni che sono svolte dall'applicazione, determinando, per esempio, in quale sequenza devono essere eseguite, quali sono le alternative, quali dati sono utilizzati, quando è richiesto un input dell'utente, come reagire a situazioni di errore.

Per lo sviluppo della piattaforma SAPI, si è scelto di modellare concettualmente il workflow di composizione di un servizio attraverso una specifica formale espressa da un diagramma di tipo grafo, in cui ogni nodo rappresenta un'attività funzionale ("activity"), ovvero una definizione del tipo di operazione che l'utente può effettuare quando si trova a quel passo dell'interazione. Il passaggio da un nodo ad un altro, e quindi la transizione da un'attività ad un'altra, è espressa mediante archi tra i nodi del grafo ed è regolata da apposite strutture di controllo (vedi Figura 2).

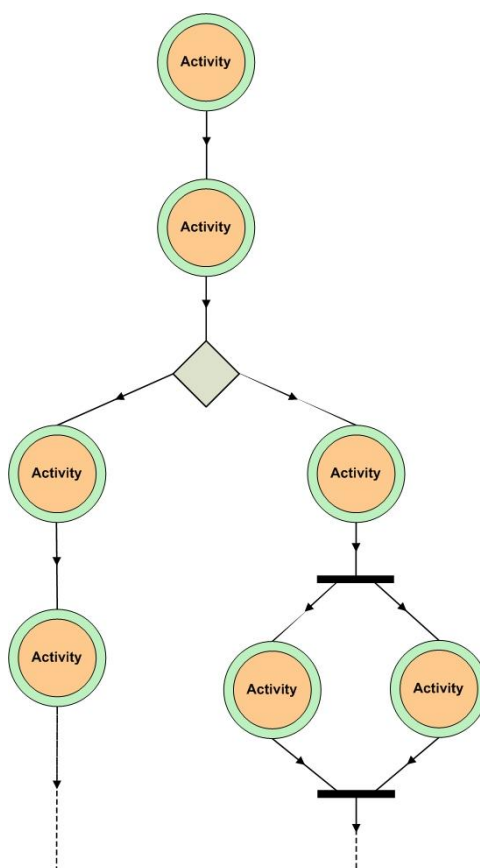


Figura 2: Esempio generico di logica di business.

Questo modello concettuale è opportunamente descritto adottando un paradigma di modellazione capace di assicurare il giusto compromesso tra genericità della modellazione e rigosità formale. Compito di questa vista del modello è quello di definire nel dettaglio tutti gli aspetti della generica activity e tutti i costrutti di composizione delle activity per formare

un workflow. Questo tipo di rappresentazione grafica è inoltre accompagnata da una descrizione tabellare che riassume gli aspetti salienti delle logiche di business sviluppate. In questa fase la specifica dei modelli non è strettamente legata ad un particolare linguaggio di implementazione essendo finalizzata all'individuazione delle informazioni fondamentali che caratterizzeranno la logica di business in SAPI. Si procede, infine, ad una opportuna traduzione utilizzando un linguaggio di descrizione semantica per la modellazione dei servizi web.

Sono previste, pertanto, due versioni del modello: una descrittiva e destinata al progettista che, a partire da essa, definirà il modello di progetto utile a chi implementa le soluzioni; una rappresentata mediante un'ontologia che servirà per vagliare le possibilità di giungere ad una ricomposizione automatica delle LdB dei servizi SAPI e che quindi sarà data in input ai motori inferenziali. La descrizione così ottenuta rappresenta uno *schema funzionale* di un e-service SAPI che contiene tutte le informazioni indicative di cosa un servizio fa, a prescindere dal come esso sia realizzato. Questa descrizione sarà, quindi, necessariamente completata (nell'attività di ricerca prevista nell'OR5) con una di tipo implementativo (*schema di deployment*), strettamente connessa alla tecnologia di deployment adottata, che descriverà il servizio in termini di entità software e sarà definita nelle fasi iniziali di sviluppo dei prototipi. In particolare, a questa specifica sarà demandato anche il compito della descrizione dei meccanismi di gestione delle eccezioni attraverso opportuni meccanismi di compensazione e la definizione di caratteristiche non funzionali.

Il mattone fondamentale nella definizione di una logica di business associata ad un servizio web erogato da SAPI, è costituito dalla activity. Nella sua essenzialità una activity può essere considerata come un processo che produce una trasformazione dei dati ricevuti in input, e causa la transizione del mondo circostante da una condizione ad un'altra. Ogni activity ha quindi associato un insieme di input e un insieme di output, che rappresentano la trasformazione dei dati provocata dal processo, e un insieme di precondizioni ed effetti che rappresentano i requisiti necessari all'esecuzione del processo e i cambiamenti provocati nel mondo esterno. In particolare, in SAPI sono previsti due tipi fondamentali di activity: atomica e composta.

Una activity atomica rappresenta un processo atomico che non può essere ulteriormente decomposto (non ha sottoprocessi) ed è eseguito in un solo passo: ad un gruppo di input, il processo atomico risponde con uno o più output, senza che l'utente abbia alcuna visibilità dell'esecuzione interna al processo. Il modello di una activity atomica è costituito, vedi Figura 3, da una descrizione del comportamento di Input/Output (Modello I/O) e del comportamento funzionale in termini di Precondizioni/Postcondizioni e Operazioni svolte dall'activity (Modello Funzionale).

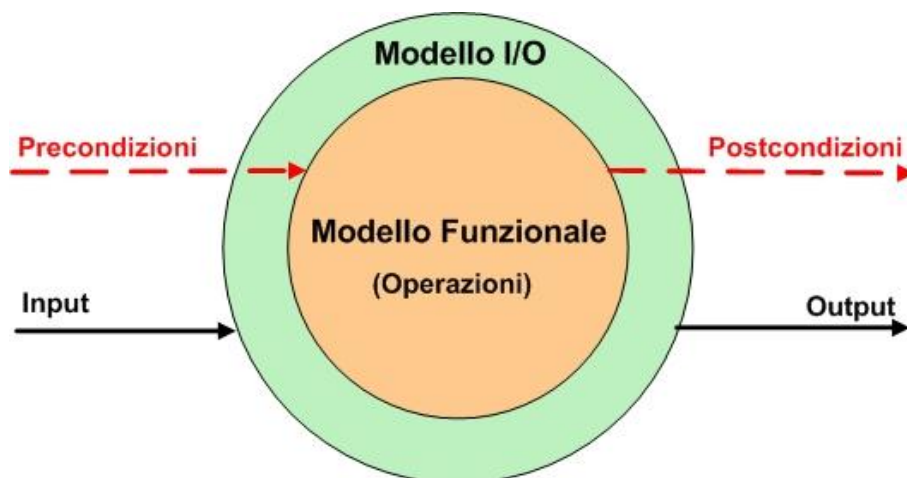


Figura 3: Activity atomica.

Le informazioni contenute nel Modello Funzionale di una activity atomica sono:

- **Nome Activity:** nome generico associato all'activity atomica.
- **ID:** identificativo univoco dell'activity atomica.
- **Descrizione:** descrizione di carattere generale del profilo funzionale esposto da una activity atomica. Questa descrizione rappresenta una vista "esterna" dell'attività che viene schematizzata come una scatola chiusa che esibisce un certo comportamento.
- **Body:** descrizione dei passi da compiere per l'esecuzione della singola attività di business. Questa descrizione rappresenta una vista "interna", ma non ancora implementativa, dell'attività che indica la sequenza di azioni da compiere ed eventuali vincoli sull'ordine di esecuzione. In questo ambito per azioni indichiamo funzionalità dell'activity atomiche, definite ad un livello di dettaglio fine, che non sono direttamente esposte ma che, nel loro complesso, ne determinano il comportamento funzionale. Nella descrizione dei passi procedurali possono essere legate, ad ogni operazione, le relative variabili di input/output definite nel modello I/O. Le informazioni contenute nel body non saranno rappresentate semanticamente ma serviranno come punto di partenza per la costruzione dello schema di deployment associato all'activity atomica.
- **Precondizioni:** condizioni di business che devono essere verificate affinché il servizio possa avere luogo. In queste condizioni non sono incluse quelle relative agli eventi generati dall'utente che innescano l'inizio del processo.
- **Postcondizioni:** risultati dell'esecuzione di una activity che comportano cambiamenti mappabili direttamente sul dominio di business definito.

Le informazioni contenute in un Modello I/O di una activity atomica sono:

- **Nome Variabile:** nome generico, indicato in corsivo, associato al parametro di scambio.

- **Descrizione:** descrizione di carattere generale del dato dell'activity atomica. Questa descrizione può eventualmente contenere indicazioni circa il tipo di dato, i valori ammissibili, le operazioni consentite ecc.
- **Entità Logica in Relazione (ELR):** i dati associati ad una activity possono essere scambiati con tre differenti entità logiche del framework SAPI (vedi Figura 4). Le possibili entità logiche sono:
 - *Activity:* sono quei dati di scambio con le altre activity del servizio collegate direttamente a quella in esame.
 - *Back-End:* sono quei dati di scambio con i repository del back-end aziendale. Questa tipologia di dati è pensata espressamente per facilitare la definizione dello schema di deployment associato all'activity atomica.
 - *Entity:* sono quei dati frutto dell'interazione diretta con l'utente, prodotti ad esempio dalla compilazione di form o destinati ad essere presentati a schermo. Come si può osservare in Figura 4, lo scambio di dati con una entity è, in pratica, finalizzato a permetter un dialogo logico tra activity e AIU (linea rossa tratteggiata). Per consentire la voluta separazione dei livelli di composizione di un servizio la modellazione del workflow di un servizio non farà, infatti, alcuna assunzione né sulle modalità di presentazione dei contenuti né sul design dell'interfaccia utente. Questi livelli non comunicheranno direttamente tra di loro, ma si sincronizzeranno solo sulla base dei dati, contenuti nelle entity, strettamente legati all'interazione con l'utente e necessari all'avanzamento della logica di business.
- **Input/Output:** indica se il dato è di input o di output.
- **Mandatory:** indica se il dato è obbligatorio o meno.
- **Tipologia:** attributo che indica la tipologia possibile del dato. In SAPI è prevista una distinzione dei dati legati ad una activity sulla base del tipo di utilizzo previsto. Le tipologie di dato previste sono:
 - *Elaborazione:* sono i dati propriamente detti, ovvero quelli usati dalle operazioni compiute dall'activity.
 - *Scelta Utente:* sono quei dati che codificano gli eventi generati dall'utente che possono innescare l'inizio del processo di business atomico o determinarne, eventualmente, la fine.
 - *Controllo o Condizione di Uscita:* sono quei dati di output che servono nel caso di impiego della activity atomica a formare workflow di activity composte per la risoluzione di eventuali branching.

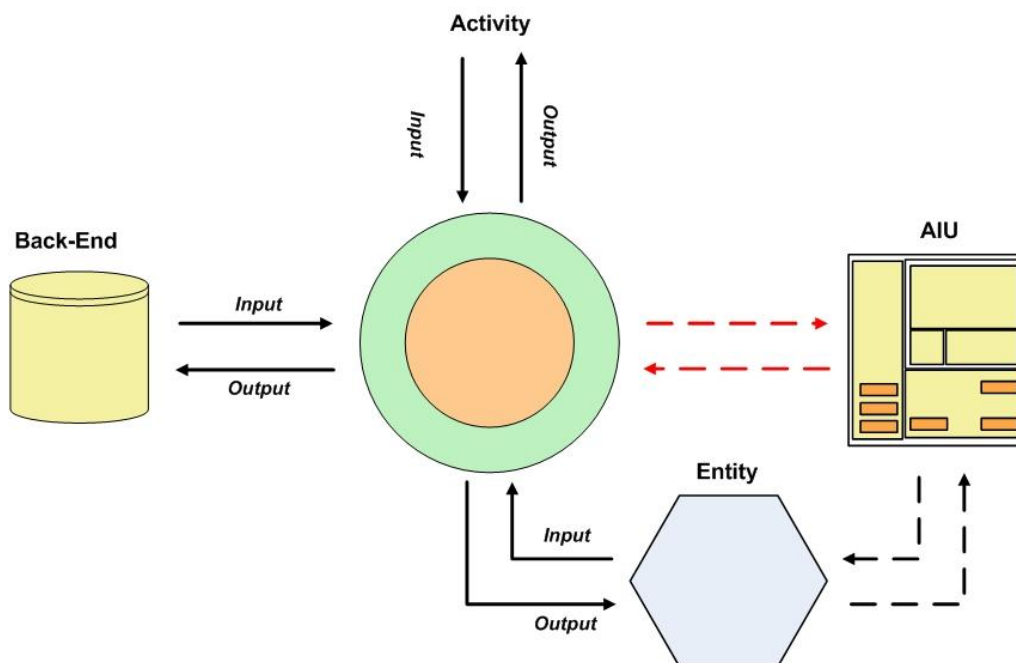


Figura 4: Legami I/O di una activity atomica.

L'activity composta, invece, è costituita da altri sotto-processi, ovvero da altre activity che possono essere sia composte che atomiche. Una activity composta permette una visione ad alta granularità dell'esecuzione di un servizio offrendo così la possibilità di avere una visione "black box" del processo. Anche le activity composte possono essere annidate a piacimento; questo garantisce una grande flessibilità e può aiutare a definire transazioni complesse, più robuste e più facili da gestire in caso di errori ed eccezioni.

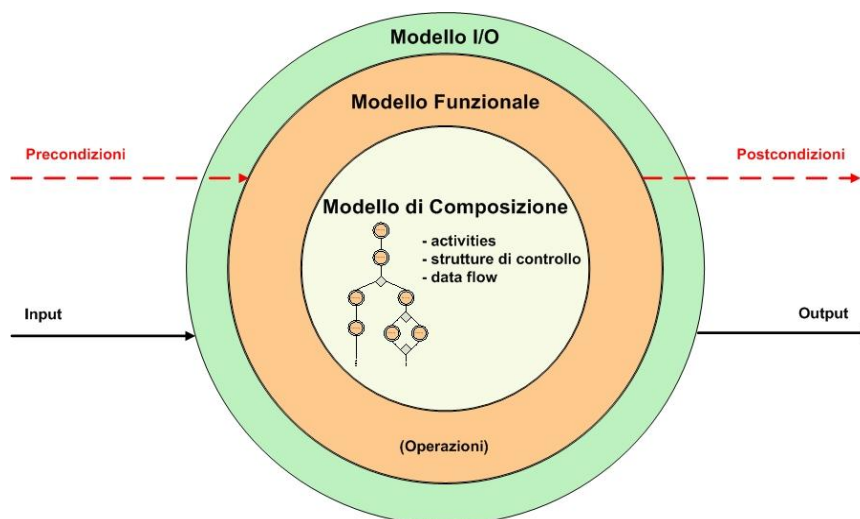


Figura 5: Activity composta.

Come mostrato in Figura 5, ad ogni activity composta è associata, in aggiunta ai Modelli I/O e Funzionale, una descrizione del workflow di composizione (Modello di Composizione), ovvero una rappresentazione delle activity che la compongono, delle strutture di controllo ad essa associate e del relativo binding dei parametri (flusso di dati e flusso di controllo).

I Modelli I/O e Funzionale associati ad una activity composta sono simili per molti aspetti a quelli associati ad una activity atomica, ma se ne differenziano per i seguenti aspetti:

- Il Modello Funzionale è sprovvisto del "Body". Questa scelta è giustificata dalla volontà di eliminare inutili ridondanze, infatti, le informazioni di un body di una attività composta sono immediatamente desumibili a partire dal Modello di Composizione e dalla descrizione funzionale di tutte le activity figlie. Inoltre per l'ID di una activity composta si possono usare solo caratteri maiuscoli.
- Il Modello I/O di una activity composta presenta sostanziali differenze rispetto a quello di una activity atomica. Per questo modello non sono presenti i dati di tipo Back End ed Entity dato che il loro utilizzo è fortemente connesso all'implementazione delle singole activity atomiche. In compenso, il modello I/O di una activity composta è stato ampliato con una nuova tipologia di parametri di scambio, i *Segnali di Decisione (SD)*, che sono quei segnali, generati dal business (corrispondono ai Segnali di Controllo¹ di una activity atomica) a partire, tipicamente, dalle scelte esplicite dell'utente (SU), che servono per risolvere i punti di decisione di un workflow, ovvero quei punti della logica di business in cui il controllo segue percorsi diversi a seconda della condizione associata. Questa tipologia di parametri è presente solo nel Modello I/O di una activity composta in quanto il suo utilizzo è circoscritto alla navigazione di un grafo di composizione.

Un'ultima differenza consiste nella presenza di un nuovo valore nel campo I/O. Infatti una activity composta oltre a dati di input ed output può avere anche dati di tipo *Local (L)*, ovvero dati di scambio tra sue sottoattività.

La scelta di mantenere un modello funzionale identico per i due tipi di activity, atomica e composta, consente di dare una visione "ricorsiva" della composizione di e-service. Infatti, il processo definito come insieme di servizi collaboranti, può essere esso stesso un servizio. Questo nuovo servizio "a valore aggiunto", così creato, potrebbe essere visibile all'esterno dal punto di vista delle funzionalità esposte, come un servizio semplice di cui è dato uno schema funzionale.

La sola grossa differenza tra una activity atomica ed una composta è, quindi, costituita dal Modello di Composizione che racchiude tutte le informazioni necessarie all'orchestrazione di un insieme di activity. Le informazioni contenute in un Modello di Composizione sono:

- **Elenco activity componenti:** elenco delle activity figlie.
- **Struttura di composizione:** una struttura di composizione è una sequenza di triple

Activity sorgente – Struttura di Controllo → (Insieme di activity destinazione)

A seconda del tipo di struttura di controllo l'insieme di destinazione può essere composto da una sola activity o da una "bag" di activity che possono essere eseguite in parallelo o in maniera alternativa.

Le principali strutture di controllo che verranno utilizzate sono:

- **Sequenza:** rappresenta un collegamento diretto tra due activity che devono essere eseguite rigorosamente in ordine.

¹ Per semplificare la modellazione di un workflow si adatterà la convenzione che i segnali SU sono sempre interpretati da AA di scelta che generano segnali di controllo.

- **Fork:** rappresenta una divisione di un cammino in due o più cammini paralleli.
- **Join:** rappresenta una congiunzione di due o più cammini paralleli in uno (barriera di sincronizzazione).
- **Branch:** rappresenta un collegamento ad una lista di activity eseguibili dalla quale ne sarà scelta una.
- **Merge:** punto in cui due o più percorsi alternativi confluiscono in un unico flusso.
- **Binding segnali di decisione:** elenco costituito dalle coppie

Decision_Var = segnale di decisione
(segnale di controllo. activity di scelta) == segnale di decisione

che associano ad ognuna della variabili di decisione del workflow il corrispettivo segnale di decisione.

- **Binding parametri I/O:** elenco costituito dalle coppie

[(nome variabile O, ID activity sorgente) ; (nome variabile I, ID activity utilizzatrice)]

La semantica di questo binding esprime, quindi, il collegamento tra un parametro di output di una activity, detta sorgente, ed il corrispettivo di input dell'activity seguente, detta utilizzatrice, sulla base della coppia nome variabile - ID activity. Nel caso in cui il parametro sia scambiato tra le sotto-attività e l'activity composta stessa, il campo ID activity sorgente o destinazione sarà riempito con la parola chiave "THIS".

Per poter riassumere in maniera sintetica ed immediata le caratteristiche di una activity composta, ed in particolar modo quelle contenute nel modello di composizione, si è scelto di adottare un formalismo grafico, basato sugli activity diagram UML.

Una volta specificato in questo modo il modello del business, si vaglierà la possibilità di una composizione, in maniera automatica o semi-automatica, di un servizio intelligente andando ad integrare opportunamente le informazioni contenute in esso con quelle riguardanti la logica di adattamento dei contenuti e di presentazione degli stessi. Uno degli aspetti più interessanti legati alla definizione della logica di business di un servizio intelligente è infatti rappresentato dalla volontà di esplorare un approccio innovativo per giungere alla definizione del workflow dei servizi erogati. Si vaglieranno le possibilità offerte dalla descrizione semantica per consentire una ricomposizione dinamica del workflow di un servizio sulla base del rapporto tra goal associati al modello utente e funzionalità esposte dalle activity definite in fase di progetto. La modellazione semantica di un e-service sembra infatti essere la strada più promettente per arrivare, attraverso l'utilizzo di appositi strumenti di reasoning, ad una ricomposizione automatica della logica di business capace di far fronte, da un lato, alle esigenze del richiedente il servizio e, dall'altro, alle risorse disponibili. A tal fine si procederà ad una traduzione semantica dei concetti contenuti nei modelli definiti in precedenza. Lo standard di modellazione ontologica adottato, a valle dell'analisi svolta, è OWL-S.

Per effettuare la voluta trasposizione si è scelto di procedere con un approccio conservativo teso a mantenere inalterata la struttura ontologica di un Service Model OWL-S ed estenderla per inglobare le caratteristiche del modello di logica di business che non possono essere espresse dalle classi e relazioni definite nello standard. La specifica semantica del workflow di composizione di un servizio SAPI, a partire dallo standard OWL-S, ha tenuto conto, quindi, di esigenze contrastanti, riuscendo a garantire un giusto trade-off tra il vantaggio costituito dall'adozione di uno standard e le inevitabili limitazioni rappresentate da un'adozione troppo rigida di un modello già strutturato.