

**Stai sviluppando un'applicazione per il Semantic Web e non trovi soluzioni per una corretta modellazione del dominio?**

Nel Semantic Web l'utilizzo e la gestione di conoscenza nella forma di regole logiche può rivelarsi problematica, specie nei casi in cui la modellazione del dominio di interesse richiede formalismi più espressivi di OWL (Web Ontology Language). In questi casi, SWRL (Semantic Web Rule Language) può rivelarsi un valido aiuto in quanto rappresenta una estensione del linguaggio per la rappresentazione delle ontologie nel Semantic Web.

**Che cos'è SWRL?**

SWRL è un linguaggio per esprimere regole logiche nel Semantic Web. Questo tipo di regole serve per esplicitare proprietà complesse sulle risorse presenti nel Web quando la potenza espressiva di OWL non è sufficiente. Il classico esempio è quello della definizione degli *zii di sangue* e *zii acquisiti*. La prima può esprimersi come:

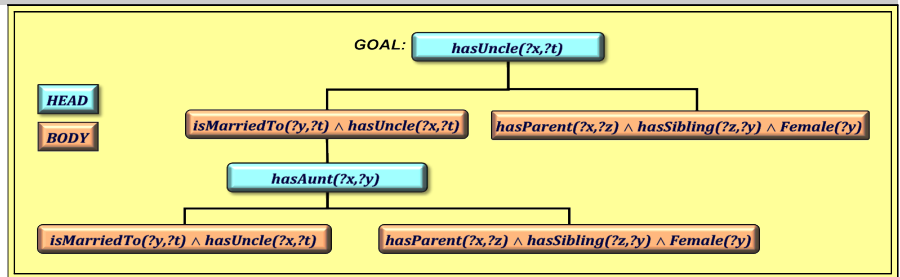
```
?x = Ivan, ?y = Clotilde, ?z = Andrea, ?t = Pietro
1. hasParent(?x,?z) ^ hasSibling(?z,?y) ^ Female(?y) -> hasAunt(?x,?y)
2. isMarriedTo(?y,?t) ^ hasUncle(?x,?t) -> hasAunt(?x,?y)
3. isMarriedTo(?y,?t) ^ hasAunt(?x,?y) -> hasUncle(?x,?t)
4. hasSibling(?y,?z) ^ Male(?z) ^ hasChild(?y,?x) -> hasUncle(?x,?z)
* namespace: http://www.abrain.it/myfamilyrules
```

◆ SE "Clotilde è sposata con Pietro" E "Ivan ha zio Pietro" ALLORA "Ivan ha zia Clotilde"

La seconda come

◆ SE "Ivan ha genitore Andrea" E "Andrea ha sorella Clotilde" ALLORA "Ivan ha zia Clotilde"

Un composer di regole SWRL (Semantic Web Rule Language) che dato un goal specificato come una classe o una proprietà OWL (Web Ontology Language) e l'insieme di regole SWRL da considerare, costruisce un albero avente come radice il goal e come rami sotto-insiemi di regole innestate. Integra il controllo "safety" (solo le variabili presenti nell'antecedente di una regola possono apparire nel conseguente) e sull'applicabilità delle regole chiamato "DL-safe" (le regole sono applicabili solo per gli individui che appaiono nella base di conoscenza). Inoltre, il composer è indipendente dal nome delle variabili, queste possono avere nomi differenti pur mantenendo la stessa semantica (effettua il controllo a livello del concetto ontologico a cui le variabili fanno riferimento).



**Caratteristiche**

Compos Br@in, l'unico tool basato su SWRL presente sul mercato in grado di rispondere alle esigenze relative alla creazione automatica di processi o workflow costituiti da attività che poggiano su descrizioni ontologiche del Semantic Web. Con Compos Br@in è sufficiente mappare in SWRL le classiche condizioni necessarie per l'esecuzione delle attività per ottenere l'albero delle composizioni per ognuno dei Goal considerati. Caratteristica importante è l'interoperabilità semantica, infatti durante il processo di composizione sono considerati eventuali allineamenti effettuati sulle ontologie di riferimento delle regole. Per la creazione delle regole può essere utilizzato un qualunque editor di testo o un editor specifico come uno dei plugin per protégé. La costruzione automatica di Semantic Web Service o Business Process complessi sono validi esempi di applicabilità di Compos Br@in.

**Requisiti Software**

OS: Indipendente dalla piattaforma  
Java Runtime Environment\*: 1.6 32-bit o 64-bit

\* Oracle e Java® sono marchi registrati di proprietà di Oracle e/o suoi affiliati.

**Requisiti Hardware**

Il tool richiede solo una quantità di RAM inferiore a un gigabyte e un processore di velocità adeguata, come disponibile in computer desktop e laptop recenti, per l'uso standard. Ontologie particolarmente complesse possono richiedere fino a 4 gigabyte di RAM, e la velocità dipenderà dal processore usato. Al momento il sistema non fa uso di processori multicore.