

# TransPlan: Uma Ferramenta para Mapear e Planejar Serviços Web Semânticos

Marcus Vinicius A. Silva, Daniela B. Claro  
Laboratório de Sistemas Distribuídos (LASID)  
Departamento de Ciência da Computação  
Universidade Federal da Bahia  
Av. Adhemar de Barros, s/n, Ondina

## Abstract

*A Web service format has been increasingly used by companies to publish their services over the Internet. In certain situations, these services need to be combined leading to a composition. Artificial Intelligence planners have been widely studied in order to achieve this composition in an automatically way. In this context, the aim of this paper is to present a tool that eases the mapping between a semantic Web service and a specific planner called JSHOP2. Although other tools were already proposed in academia, none of them works with such HTN planner for Java. Experiments were carried out using a specific scenario so as to validate such mapping and a comparative table with related work was presented at the end of this paper.*

## 1 INTRODUÇÃO

Os serviços Web são componentes de *software* que podem ser publicados, localizados e invocados através da Web de uma forma padronizada. A descrição semântica facilita a descoberta automática destes serviços distribuídos na Internet e pode ser realizada através da linguagem *Ontology Web Language for Services* (OWL-S) [4]. Diversos pesquisadores vêm utilizando os planejadores da Inteligência Artificial para compor serviços Web automaticamente, especificamente o planejador JSHOP [1, 2, 6]. Porém, a integração destas diferentes tecnologias e consequentemente linguagens, muitas vezes não ocorre de forma automática, sendo necessário realizar uma tradução entre a linguagem semântica utilizada para descrever serviços Web e o planejador utilizado.

Diante desse contexto, este trabalho apresenta uma ferramenta, denominada TransPlan, para realizar o mapeamento entre os serviços Web descritos através da OWL-S e o planejador JSHOP2. De modo particular, pretende-se: (a) converter automaticamente a descrição semântica dos serviços Web para uma linguagem compreensível pelo planejador JSHOP2; (b) planejar a seqüência de execução dos serviços Web exibindo todas as soluções encontradas.

O presente artigo está assim organizado: a seção 2 descreve os trabalhos relacionados que serão utilizados no comparativo com a solução proposta. A

seção 3 explana uma visão geral dos serviços *web*, enfatizando a composição automática. A seção 4 apresenta os planejadores abordando a técnica HTN (*Hierarchical Task Network planning*) para composição automática de serviços *web*. A seção 5 expõe a composição automática através do TransPlan. A seção 6 mostra o cenário utilizado para testes da solução proposta e apresenta os resultados obtidos. E por fim, na seção 7 são explanadas as conclusões e os trabalhos futuros.

## 2 TRABALHOS RELACIONADOS

Os autores em [1] propõem a criação de um *workflow* científico utilizando planejadores da IA combinados com a composição de serviços manual, semi-automática e automática. A composição automática foi realizada através da utilização do planejador SHOP2.

O trabalho em [5] apresenta uma ferramenta para compor serviços Web através do planejador JSHOP, denominada OWLS2JSHOP. Os autores confirmam a pertinência da aplicação de um planejador HTN para a atividade de composição automática de serviços Web.

Os autores em [6] propõem uma ferramenta para as fases de planejamento e execução da composição de serviços Web. Especificamente sobre a conversão OWL-S para o SHOP2, desenvolvido em LISP, é proposto um mapeamento de processos atômicos e compostos para respectivamente operadores e métodos.

O TransPlan apresenta as seguintes vantagens em relação aos trabalhos relacionados: (a) o TransPlan localiza e disponibiliza os diversos planos encontrados atribuindo um custo específico a cada ação do plano. (b) o TransPlan é compatível com a versão mais atual do SHOP que é o JSHOP2, e, segundo os autores em [3], ela apresenta melhorias significativas relacionadas a desempenho. E, diferentemente do TransPlan, as demais ferramentas mencionadas acima são compatíveis exclusivamente para a utilização com um único planejador pré-determinado.

### 3 COMPOSIÇÃO AUTOMÁTICA DE SERVIÇOS WEB

A composição de serviços *web* é possível através da publicação da interface de um serviço, o que permite que suas operações sejam combinadas a fim de oferecer um novo serviço. As informações contidas nos documentos WSDL apenas descrevem sintaticamente os serviços. Algumas linguagens foram criadas com o intuito de descrever semanticamente estes serviços, como a *Ontology Web Language for Service* (OWL-S), *Web Service Modeling Ontology* (WSMO) e *Semantic Annotations for WSDL and XML Schemas* (SAWSDL). Neste trabalho foi utilizado a OWL-S para descrição semântica dos serviços por ser uma variação da OWL (Web Ontology Language) recomendada pela W3C, além de ser a linguagem utilizada nos dois cenários experimentais.

O processo automático de composição de serviços demanda a informação de um objetivo através do qual será possível determinar quais e em qual ordem os serviços deverão ser invocados (o fluxo para execução dos serviços). Uma composição automática é composta de essencialmente três fases: descoberta, planejamento e execução [5]. A fase de **descoberta** tem como objetivo localizar os serviços *web* que poderão ser utilizados para atender determinado objetivo do usuário. A fase de **planejamento** é responsável por determinar quais e em que ordem os serviços serão utilizados. E, na fase de **execução** todos os serviços identificados (na fase de planejamento) e alinhados na composição serão invocados.

O objetivo desse trabalho concentra-se na fase de planejamento. De forma análoga, o objetivo (estado final) e as execuções dos serviços (ações) podem ser mapeados como um tipo de problema resolvido pelo planejamento da inteligência artificial [7, 6, 5].

A próxima seção apresenta uma visão geral sobre os planejadores da Inteligência Artificial e a utilização desses na composição automática de serviços.

### 4 PLANEJADORES DA INTELIGÊNCIA ARTIFICIAL

Os planejadores da IA (Inteligência Artificial) surgiram de investigações em busca no espaço de estado, prova de teoremas e teoria de controle, e das necessidades práticas da robótica. Atualmente, diversas técnicas de planejamento têm sido utilizadas. Cada uma mais apropriada a um determinado tipo de problema. Dentre elas, o planejamento de Rede Hierárquica de Tarefa - HTN (*Hierarchical Task Network planning*) tem sido amplamente utilizado pela área de composição automática de serviços *web* [6], pelo fato de que cada nível da hierarquia é reduzido a um número menor de atividades até encontrar atividades atômicas (operadores). Os autores em [6] apresentam diversas características que justifica a utilização da técnica HTN na composição de serviços *web*.

O planejador SHOP e seus sucessores implementam uma variação da técnica HTN também denomi-

nada *Simple Hierarchical Ordered Planner* (SHOP). Esse planejador classificou-se entre os 4 dos 14 planejadores que concorreram na competição internacional de planejamento [6]. A versão JSHOP2 - *Java Simple Hierarchical Ordered Planner* destaca-se em relação às versões anteriores devido a otimização de desempenho [3], motivo pelo qual é utilizado neste trabalho.

Os planejadores SHOP2 planejam as tarefas na mesma ordem em que serão posteriormente executadas. Essa característica permite conhecer o estado atual do mundo em cada etapa do processo de planejamento, o que torna possível a avaliação de mecanismos para inferência, incluindo a capacidade para chamar programas externos. Isto é apontado por [6] como relevante por permitir integrar o planejamento com informações externas, como no ambiente *web*.

### 5 A FERRAMENTA TRANSPLAN

A ferramenta TransPlan é uma solução *open-source*, distribuída sob licença LGPL (*Lesser General Public License*), cujo objetivo é realizar o mapeamento entre os serviços Web descritos em OWL-S e o planejador JSHOP2. A Figura 1 apresenta a arquitetura de alto nível do TransPlan, destacando os seus principais módulos lógicos.

- **Planner.** Esse componente é responsável por invocar o planejador configurado no TransPlan, que no contexto desse trabalho, corresponde ao JSHOP2. Além disso, o TransPlan gera a partir da OWL-S o domínio e o problema necessários para que o JSHOP2 possa iniciar o planejamento.
- **OWL-S Reader.** Esse componente é responsável por carregar uma OWL-S a partir de sua *Uniform Resource Identifier* (URI). A OWL-S API utilizada na solução é a disponibilizada pela MINDSWAP [6].

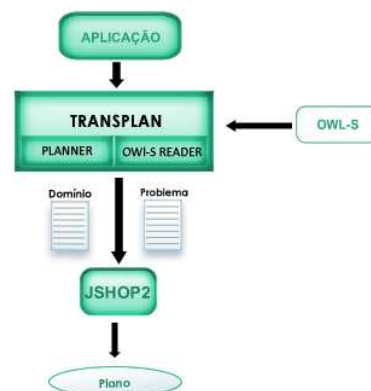


Figure 1: Arquitetura de alto nível do TransPlan

A Figura 2 apresenta um dos algoritmos implementados no TransPlan. O algoritmo exibido converte um arquivo em OWL-S para JSHOP2. A implementação do algoritmo levou em consideração que

- Entrada:** uma definição OWL-S de um processo atômico (A)  
**Saída:** um operador JSHOP2 (O)
1. Para cada pré-condição (p) em A
    - 1.1. Adicionar p como uma pré-condição em (O)  
 $Pre := Pre \cup \{p\}$
    - 1.2. Adicionar p como um efeito negativo em (O)  
 $Del := Del \cup \{p\}$
  2. Se a lista de pré-condição for vazia (Se  $Pre == \{\}$ )
    - 2.1. Para cada entrada(i) em A
      - 2.1.1. Adicionar (i) como uma pré-condição em (O)  
 $Pre := Pre \cup \{i\}$
      - 2.1.2. Adicionar (i) como um efeito negativo em (O)  
 $Del := Del \cup \{i\}$
  3. Para cada efeito (e) em A
    - 3.1. Adicionar (e) como um efeito positivo em (O)  
 $Add := Add \cup \{e\}$
  4. Se a lista de efeitos (Add) for vazia (Se  $Add == \{\}$ )
    - 4.1. Para cada *output*(o) em A
      - 4.1.1. Adicionar (o) como um efeito em (O)  
 $Add := Add \cup \{o\}$
  5. Retornar (:O Pre Del Add)

Figure 2: Converter processos atômicos em operadores

o JSHOP2 não faz distinção entre condições, com origem em entradas e saídas de informações pelo usuário, e condições e efeitos com origem em regras do domínio [5]. Entretanto, para que o algoritmo não se limitasse a ler condições e efeitos em OWL-S, foi acrescentado uma rotina que permite mapear *inputs* para condições e *outputs* para efeitos, caso não existam condições e efeitos na OWL-S.

A Figura 3 apresenta a tela principal do TransPlan quando executado por um usuário em ambiente *Desktop*. Através dessa tela é possível carregar a OWL-S, definir o estado inicial, o objetivo e executar o planejador configurado. O TransPlan também pode ser reutilizado no desenvolvimento de aplicações deste mesmo domínio, permitindo a abstração da etapa de planejamento, já implementada pela solução.

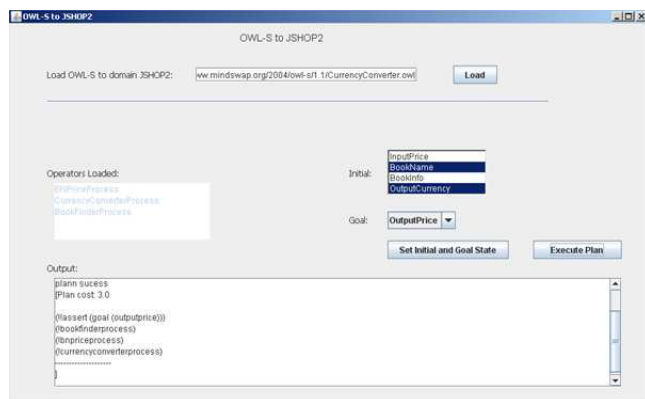


Figure 3: Tela Principal do TransPlan

## 6 EXPERIMENTOS REALIZADOS

A ferramenta proposta foi avaliada através da utilização de um cenário: busca e conversão de preço de livros (*BookFinder*) utilizado em [5]. Neste cenário foram realizadas as seguintes validações: (a) Carregar ontologias OWL-S com o intuito de testar o módulo que manipula ontologias OWL-S, (b) apresentar o plano para cada um dos cenários apresenta-

dos fornecendo estado inicial e objetivo válidos e (c) inserir entradas inválidas, não informando nem os estados iniciais e nem o objetivo a fim de verificar se dados incorretos continuam gerando um plano válido.

### 6.1 BUSCA E CONVERSÃO DE PREÇO DE LIVROS (*Book Finder*)

Este exemplo é uma adaptação do exemplo utilizado por [5]. Ele descreve um domínio de serviços de busca e conversão de preço de livros, a partir do nome do livro, e do tipo de moeda e retorna como resultado o preço. A Figura 4 apresenta a solução composta por três serviços em OWL-S: *BookFinderProcess*, *BNPriceProcess* e *CurrencyConverterProcess* descritos a seguir:

- *BookFinderProcess*: retorna o ISBN (*BookInfo*) de um livro sendo informado seu título (*BookName*);
- *BNPriceProcess*: retorna o preço de um livro (*BookPrice*) dado seu ISBN (*BookInfo*);
- *CurrencyConverterProcess*: converte um preço (*InputPrice*) informado em dólares, para um preço (*OutputPrice*) especificado em outra moeda (*OutputCurrency*).

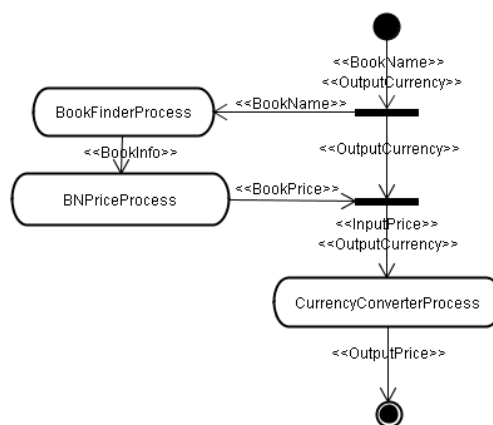


Figure 4: Solução do Cenário *BookFinder*

*InputPrice* e *BookPrice* possuem os mesmos valores semânticos na OWL-S, porém sintaticamente estes são distintos. A ferramenta TransPlan permite definir axiomas com o intuito de mapear estas divergências sintáticas e permitir a composição semântica dos serviços OWL-S envolvidos na troca de mensagens.

### 6.2 ANÁLISE DOS RESULTADOS

Para análise dos resultados foi realizado um estudo comparativo entre os trabalhos relacionados (seção 2) e o TransPlan. A Tabela 1 apresenta as características específicas de cada ferramenta: o OWL2JSHOP[5], o JSHOPTranslator[6] e a ferramenta proposta o TransPlan. A seguir são apresentados os critérios analisados.

- 1. Suporte a planejador genérico.** Essa característica permite o desacoplamento do planejador utilizado. Como não existe um consenso com relação ao planejador ideal para a composição de serviços Web, este critério torna-se relevante na modularização da solução.
- 2. Abstração no Planejamento.** Abstrair do usuário o conhecimento da linguagem utilizada pelo planejador, permitindo que, de posse das descrições semânticas do serviços incluindo os estados inicial e final, seja possível localizar as soluções disponíveis automaticamente, sem a intervenção do usuário.
- 3. Geração automática de método para decomposição.** A geração automática do método para decomposição em busca do plano permite que somente seja necessário carregar os processos atômicos da OWL-S. O TransPlan atende parcialmente esse requisito pois limita-se a utilizar a estrutura IF-Then-Else.
- 4. Definição Semântica automática.** Essa característica permite converter os tipos semânticos, definidos em OWL-S, automaticamente para a linguagem utilizada pelo planejador. No TransPlan essa característica foi contemplada parcialmente visto que é necessária a intervenção do usuário na definição dos axiomas.
- 5. Etapas da composição.** As fases da composição incluem descoberta, planejamento e execução. O TransPlan limita-se a fase de planejamento.
- 6. Uso de IOPE.** O conjunto entrada (I), saída (O), condição (P) e efeito (E) deverá ser utilizado pelo planejador para encontrar a solução.
- 7. Compatibilidade com o JSHOP2.** O JSHOP2 é a versão mais atual do planejador SHOP2 para Java.

Table 1: Comparação com os trabalhos relacionados

CARACTERÍSTICAS	TRANSPLAN	OWL2JSHOP	JSHOPTranslator
1. Suporte a planejador genérico	Atende	Não atende	Não atende
2. Abstração no Planejamento	Atende	Não mencionado	Parcialmente
3. Geração automática de método para decomposição	Parcialmente. Utiliza IF-then-else	Não Atende. Lê definição em OWL-S	Não Atende. Lê definição em OWL-S
4. Definição Semântica automática	Parcialmente	Atende	Atende
5. Etapas da composição	Parcialmente. Apenas Planejamento	Parcialmente. Apenas planejamento.	Parcialmente. Planejamento e execução
6. Uso de IOPE	Atende	Parcialmente. Apenas I/O	Parcialmente Apenas I/O
7. Compatibilidade com o JSHOP2	Atende	Não Atende	Não Atende

## 7 CONCLUSÃO

Este trabalho propôs uma ferramenta para mapear os serviços Web semânticos descritos em OWL-S e o planejador JSHOP2. Duas análises foram realizadas com o intuito de validar a ferramenta proposta: a) execução de dois cenários de testes e b)

utilização de uma tabela comparativa com trabalhos relacionados. O TransPlan oferece algumas características relevantes para o mapeamento de serviços em OWL-S para o JSHOP2. E ainda, o TransPlan se destaca pela sua flexibilidade para extensão e customização, por exemplo, para utilização com outros planejadores. Como trabalho futuro, é necessário incorporar as outras fases do planejamento com o intuito de oferecer uma ferramenta integrada para a composição automática de serviços Web. Instruções adicionais e acesso a ferramenta encontra-se disponível em <http://transplan.wiki.sourceforge.net>.

## AGRADECIMENTOS

Agradeço ao Instituto Recôncavo de Tecnologia pelo apoio oferecido a esse projeto de pesquisa.

## References

- [1] Digiampietri, L. A., Alcázar, J. J. P., and Medeiros, C. B. (2007). AI Planning in web services composition: a review of current approaches and a new solution. In VI ENIA - Proceedings of the XXVII Brazilian Computer Society Conference (CSBC2007).
- [2] Hakimpour, F., Sell, D., Cabral, L. and Domingue, J., and Motta, E. (2005). Semantic web service composition in irs-iii: The structured approach. In Seventh IEEE International
- [3] Ilghami, O. and Nau, D. (2003). A general approach to synthesize problem-specific planners. In Technical Report CS-TR-4597 e UMIACS-TR-2004-40 - University of Maryland.
- [4] Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Parsia, M. P. B., Payne, T., Sirin, E., Srinivasan, N., and Sycara, K. (2004). OWL-S: Semantic markup for web services. Disponível em: <http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/>. Acesso em: 5 Jun. 2006.
- [5] Murtagh, D. (2004). Automated Web Service Composition. University of Dublin. 2004. Disponível em: <https://www.cs.tcd.ie/publications/tech-reports/tr-index.05.php>. Acesso em: 15 out 2007.
- [6] Sirin, E., Parsia, B., Wu, D., Hendler, J., and Nau, D. (2004). Htn planning for web service composition using shop2. In Journal of Web Semantics.
- [7] Klusch, M., Gerber, A., and Schmidt, M. (2005). Semantic web service composition planning with owls-xplan. In Symposium on Agents and the Semantic Web.