

# Matching Web Entities with Potential Actions

Ioannis Stavrakantonakis, Anna Fensel, Dieter Fensel

University of Innsbruck, STI Innsbruck  
Technikerstr. 21a, 6020 Innsbruck, Austria  
`{firstname.lastname}@sti2.at`

**Abstract.** The creation of schema.org as the de facto vocabulary for the implementation of Semantic Annotations was the dawn of a new era for the Web by motivating the Web developers to start weaving semantics in the content, mainly, for visibility reasons in search engine results. Moving further, the new version of the vocabulary enables Web entities to self-describe the *Actions* with which they interact with users, agents or services. In this scope, we present our ongoing work on automatic weaving of *Actions* based on the existing semantic annotations of a website.

**Keywords:** Semantic Annotations, schema.org, actions, Web APIs

## 1 Introduction

The idea to build web agents to understand web content and interact with it in order to realise a given plan and achieve goals has been part of the Semantic Web vision [1] since the very first steps of the related working groups. As described by Hendler [2] with the example of intelligent travel agents, those systems should be *communicative, capable, autonomous* and *adaptive*. The various vocabularies and ontologies, that have been developed until today, contribute towards making web content machine-understandable through semantic annotations, which will enable web agents to behave as described before. The schema.org initiative aimed to accelerate the adoption and creation of semantic annotations and play the role of the de facto vocabulary in this area. An example of applying semantic annotations to an already existing infrastructure is demonstrated in [3], in a real world scenario of the touristic sector.

In the same direction, but from another perspective, the discovery of Web Services [4] aimed to build a layer that would enable the web agents to be *capable* of interacting with web entities. In this direction, the very recent extension of the schema.org vocabulary [5] brings these ideas from the research initiatives to the realisation layer of the Web by allowing entities to have a self-explanatory presentation of the actions they enable and how these actions can be invoked. In this scope, our approach aims to facilitate the uptake of the vocabulary as well as the scalability of the semantic annotations implementation.

The remainder of the paper describes the structure of the actions dimension of schema.org and highlights the difficulties of using it in section 2; outlines the ongoing work of the proposed approach in section 3; and closes with our conclusions and the current outlook in section 4.

## 2 Background

One of the great features of the new extension is the flexibility of combining any *Action* to any type of object/entity of the vocabulary. This is allowed by design as the related property has been hooked to the most generic class, the *Thing*. Specifically, the action property is named *potentialAction* with domain the class *Thing* and range the class *Action*, which is at the top of the inheritance hierarchy of the actions. Thinking in triples, the schema has been designed in such a way that more than one triple could be created in order to enable a fine-grained description of the action, participating agents, objects, results of it and time variables.

For example, a *Hotel* object could participate in many different actions as shown in Fig. 1.

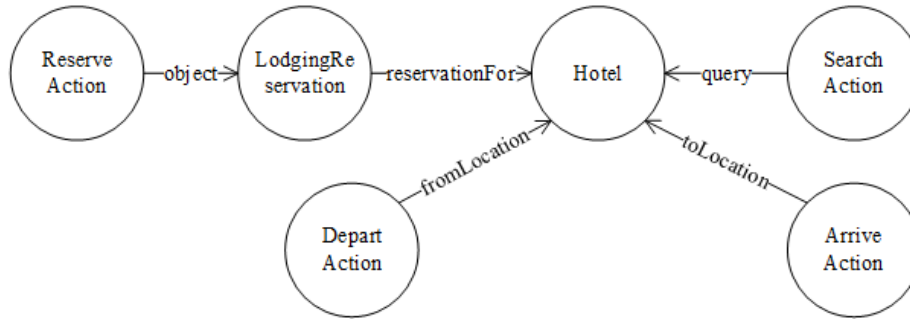


Fig. 1. Hotel class interaction with schema.org actions.

The above described flexibility enables developers to build very descriptive websites and web applications. On the other hand, it needs deep understanding of the actions hierarchy for matching the two sides in order to make the right decisions that would result in meaningfully described content with a high potential of usability and visibility. Our ongoing work aims to tackle those obstacles in the scope of a broader idea of leveraging websites to open self-described APIs by annotating the content and the provided services.

## 3 Our approach

The approach that we follow aims to address the difficulties of developing *Actions* on websites and web applications. The added value of our contribution is the generation of suggestions about potential actions with which the described web entities of a website could support and interact. We define the problem as follows. We assume that semantic annotations have already been implemented for a given website, which contains entities that would be meaningful to interact with other entities in the Web sphere. For example, in case of lodging businesses, the

offered services (e.g. reserve rooms) could be potential objects that an agent would be interested to interact beyond a simple read-only consumption (e.g. to make/cancel or update a reservation). Therefore, let  $W = \{E_1, E_2, \dots, E_n\}$  with  $n \in \mathbb{N}$  be the set of all entities described in a website. Each entity  $E \in W$  is the subject for properties such that  $E = P_T \cup P_A$ , where  $P_T$  stands for properties derived from any *class* and  $P_A$  for those that refer to the *action* class. We make this artificial distinction between properties in order to explain better our approach.

Furthermore, let  $T = \{T_1, T_2, \dots, T_t\}$  with  $t \in \mathbb{N}$  be the set of all entity classes in schema.org and  $A = \{A_1, A_2, \dots, A_m\}$  with  $m \in \mathbb{N}$  be the set of all action classes in schema.org. We define the function  $\phi : T \rightarrow \mathcal{P}(A)$  that maps each entity to the set of actions that could be used by the entity type. Thus, having specified these mappings, we are able to define an entity  $E$  and its properties  $P_T$  and  $P_A$ . The core of our approach consists of the idea that we could define potential actions for a class  $T$ , by studying the range of the properties in an *Action* class  $A$ . In the scope of this paper, we identify two possible cases. The first case, which is the simplest one, refers to the direct mapping of the range of  $P_A$  to classes  $T$ . For example, according to Fig.1, the *DepartAction*  $A_1 \in A$  has the property *fromLocation*  $P_{A1} \in P_A$  with range *Place*  $T_1 \in T$ . Thus, any subclass of  $T_1$  is in the range of  $P_{A1}$  and we could assume the  $A_1$  as a potential action for the class *Hotel* (*Place*  $\rightarrow$  *Hotel*).

In the second case, we assume that when properties  $P_A$  of a class  $A_1 \in A$  and properties  $P_T$  of a class  $T_1 \in T$ , and  $P_T$  are not derived from superclasses (e.g. the *Thing*), have matching ranges, then  $A_1$  is a potential action for  $T_1$ . Building on the example of Fig.1 we apply the above idea as shown in Table 1. In particular, the *LodgingReservation* class and the *ReserveAction* of the schema.org vocabulary are described in the table with all their properties, excluding any derived properties from superclasses. The former models a reservation entity of a lodging business and the latter depicts an action of reserving concrete objects like a hotel or a restaurant. The middle column of the table indicates the fields that map from the *Reservation* class to the related *Action* class. As it is shown, the range of all the properties of the *Action* class map to a subset of the *Reservation*'s properties.

**Table 1.** Mapping the *LodgingReservation* to the *ReserveAction*

| LodgingReservation |                 | maps to       | ReserveAction   |              |
|--------------------|-----------------|---------------|-----------------|--------------|
| property           | type            |               | type            | property     |
| checkinTime        | <i>DateTime</i> | $\rightarrow$ | <i>DateTime</i> | scheduleTime |
| checkoutTime       | <i>DateTime</i> | $\rightarrow$ |                 |              |
| numAdults          | <i>Number</i>   |               |                 |              |
| numChildren        | <i>Number</i>   |               |                 |              |

At the evaluation part of the described approach, we employ datasets based on entities that instantiate schema.org classes and websites collected via web

crawling in order to determine the occurrence of false positives and false negatives based on precision and recall measurements. An example of such dataset is our ongoing Linked Open Data dataset of touristic service providers in Innsbruck and its surroundings<sup>1</sup>.

## 4 Conclusion

Our approach could be summarised as an initiative towards the realisation of the paradigm that will transform websites to machine-readable data feeds, in other words, to APIs. At the core of this paradigm, lie the annotation of Web resources with a widely adopted vocabulary. The benefit of enabling the automatic extension of existing annotated entities with actions is threefold: a) supports the easy description of more content on a website, e.g. “*Last reservation for the 24th of August was made 3 minutes ago*”; b) promotes the exposure of specific potential actions of existing entities of a Website, e.g. a potential action for the *Hotel* is a *ReserveAction*; and c) facilitates the establishment of a communication channel between agents and the provided services of a Website.

In the future work, we consider the *EntryPoint* property of the *Action* class, which is specially interesting for Web agents, as it gives information about how to formulate a request and what to expect as a response. Furthermore, we aim to explore the matching of classes and actions with more ways and to integrate it with the infrastructure provided by the Hydra [6] approach, that enables the development of hypermedia-driven Web APIs.

**Acknowledgments.** This work has been partially supported by the EU projects BIG, BYTE, eFreight, FITMAN, LDBC, MSEE, PlanetData and PRELIDA, FWF project OntoHealth, as well as FFG projects OpenFridge and TourPack.

## References

1. Fensel, D., Hendler, J., Lieberman, H., Wahlster, W.: Spinning the semantic web. In IEEE Intelligent Systems (2003).
2. Hendler, J.: Is there an intelligent agent in your future? Nature, vol. 11(1999) <http://www.nature.com/nature/webmatters/agents/agents.html>
3. Toma, I., Stanciu, C.V., Fensel, A., Stavrakantonakis, I., Fensel, D.: Improving the online visibility of touristic service providers by using semantic annotations. In the 11th ESWC (2014).
4. Fensel, D., Lausen, H., Polleres, A., de Bruijn, J., Stollberg, M., Roman, D., Domingue, J.: Enabling semantic web services. Springer-Verlag Berlin Heidelberg (2007).
5. Brickley D.: Announcing Schema.org Actions. (April, 2014) <http://blog.schema.org/2014/04/announcing-schemaorg-actions.html>
6. Lanthaler, M.: Creating 3rd generation web APIs with hydra. In Proceedings of the 22nd international conference on World Wide Web companion (pp. 35-38). International World Wide Web Conferences Steering Committee (2013).

<sup>1</sup> <http://loi.sti2.at/openrdf-workbench/repositories/STI/summary>