



Machine Readable Web APIs with Schema.org Action Annotations

Umutcan Şimşek, Elias Kärle, Dieter Fensel

14th International Conference on Semantic Systems, Vienna, Austria

Outline

1. Motivation
2. Schema.org Actions as Web API Description Vocabulary
3. Methodology and Use Case
4. Ongoing Work

1. Motivation

„Headless Web“

- Graphical design of a website is losing its significance
- High quality, semantically annotated structured data is a necessity for online visibility
- Web pages primarily for robots, not humans!

<https://paul.kinlan.me/the-headless-web/>

1. Motivation

- To be prepared for the “headless web”, not only data but also web services should be machine understandable
- Early efforts were focused on SOAP services
 - they were a base for later lightweight efforts
- Mostly academic interest, poor adoption in the industry; the “chicken-egg problem” [Lanthaler & Gütl, 2011]

1. Motivation

- More recent, hypermedia focused approaches: Hydra [Lanthaler & Gütl, 2013], RestDesc [Verborgh et. al., 2012]...
- Approaches enriching existing interface description languages like OpenAPI: smartAPI[Zaveri et. al., 2017]...

1. Motivation



<http://www.schema.org/>

Yandex

Embedded into HTML:
Microdata
RDFa
JSON-LD

1. Motivation

- Schema.org: de-facto industrial standard
- Schema.org actions used by:
 - email clients (e.g. gmail)
 - search engines (e.g. google site searchbox)
- Schema.org actions from semantic web services perspective

2. Web API Annotation with Schema.org Actions

“The Web is not just about static descriptions of entities. It is about taking action on these entities.”

(a) As resource description

BuyAction may expect an object of type Offer and promise to return an entity of type Order

(b) As potential action on entities

A specific instance of an Offer on an Event may have a BuyAction that promises to return a Ticket instance

2. Web API Annotation with Schema.org Actions

```
{
  "@context": "http://schema.org/",
  "@type": "SearchAction",
  "name": "Search for hotel room offers",
  "object": {
    "@type": ["HotelRoom", "LodgingReservation"],
    "checkinTime-input": "required",
    "checkoutTime-input": "required",
    "numAdults-input": "required",
    "containedInPlace": {
      "@type": "Hotel",
      "amenityFeature": {
        "@type": "PropertyValueSpecification",
        "name-input": "optional"
      }
    }
  },
  "result": {
    "@type": "Offer",
    "name-output": "required",
    "itemOffered": {
      "@type": "HotelRoom",
      "name-output": "required",
      "occupancy": {
        "@type": "QuantitativeValue",
        "value-output": "required",
        "unitCode-output": "required"
      }
    },
    "price-output": "required",
    "priceCurrency-output": "required"
  }
}
```

- An action consists of the following:
 - type of action
 - object types
 - result types
 - input property descriptions
 - output property descriptions
 - invocation description

[Şimşek et. al., 2018]

2. Web API Annotation with Schema.org Actions

Mapping RESTful APIs to Schema.org Actions

REST Element	Schema.org Actions Element
Resource	schema:object Value
Resource Identifier	schema:urlTemplate
Resource Method	schema:Action, schema:PropertyValueSpecification and schema:httpMethod
Resource Representation and Metadata	schema:encodingType, schema:contentType

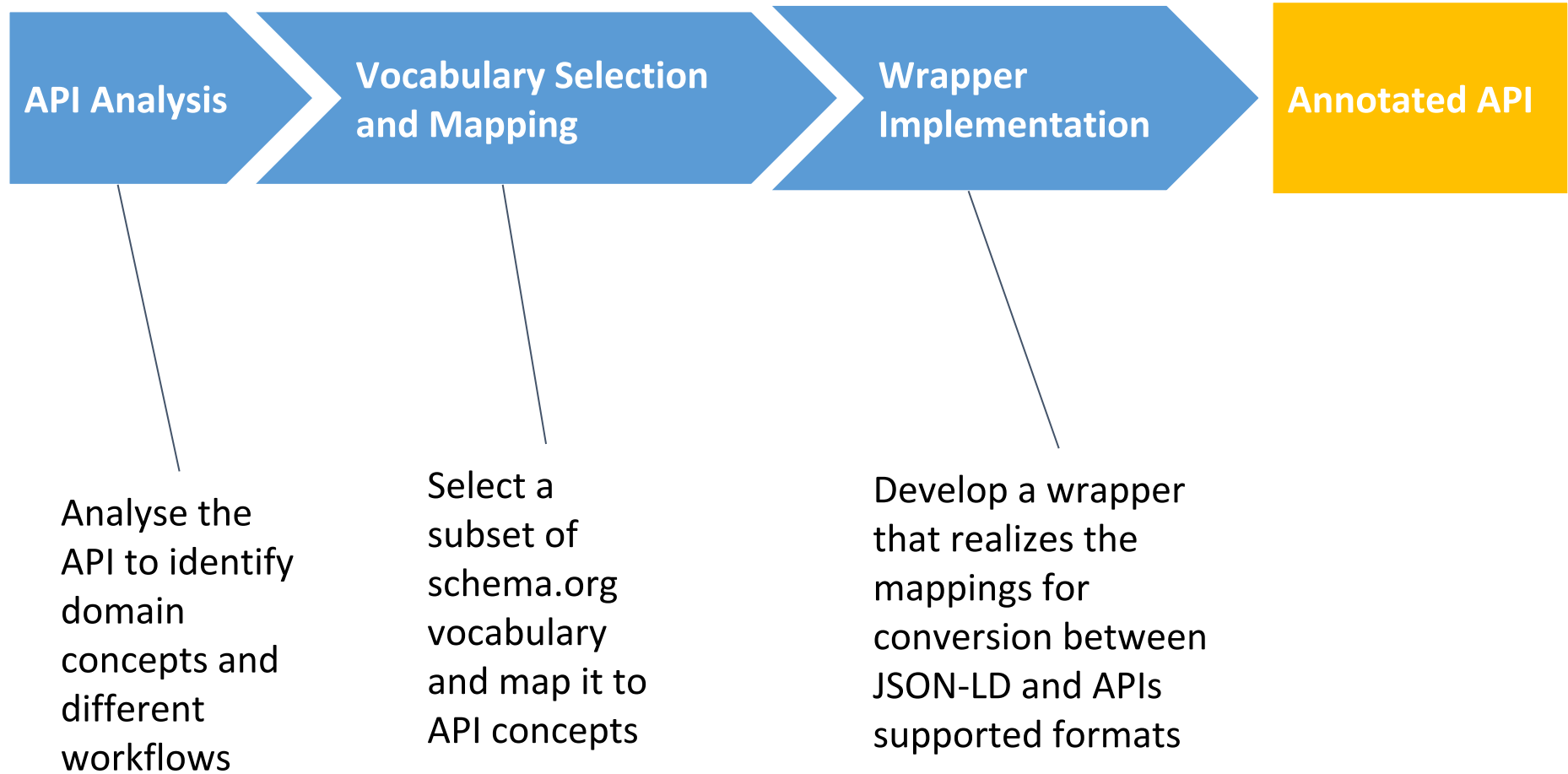
2. Web API Annotation with Schema.org Actions

- Schema.org Actions vocabulary does not have strong formal semantics
- RDFS based data model
- The semantics are partially hidden in the names and descriptions of entities

2. Service Semantics

Type of Service Semantics	Schema.org Actions Support
Information Model	<ul style="list-style-type: none">- RDFS based schema.org vocabulary and its extensions
Functional Semantics	<ul style="list-style-type: none">- Signature view with input and output definitions
Behavioural Semantics	<ul style="list-style-type: none">- Implicit with potential actions
Non-Functional Semantics	<ul style="list-style-type: none">- No support in the core- Pending WebAPI type- External authentication extension

3. Annotating Web APIs



3. Use Case: Easybooking API

GET api/searchCatAvailability/ →
schema:SearchAction

schema:object → schema:LodgingReservation

schema:result → schema:Offer

Input Parameter	Schema.org Property
arrivalDate	checkinTime
departureDate	checkoutTime
numAdults	numAdults

```
{
  "@context": "http://schema.org/",
  "@type": "SearchAction",
  "name": "Search for hotel room offers",
  "object": {
    "@type": ["LodgingReservation"],
    "checkinTime-input": "required",
    "checkoutTime-input": "required",
    "numAdults-input": "required"
  },
  "result": {
    "@type": "Offer",
    "name-output": "required",
    "itemOffered": {
      "@type": "HotelRoom",
      "name-output": "required",
      "occupancy": {
        "@type": "QuantitativeValue",
        "value-output": "required",
        "unitCode-output": "required"
      }
    },
    "price-output": "required"
  }
}
```

3. Use Case: Action Processing Bot

```
1 User: I am searching for the Hotel STIIInnsbruck.
2 Bot: One item has been found. Hotel STIIInnsbruck. You can ask me something about it.
3 User: Tell me more.
4 Bot: You can search and find hotel rooms and offers.
5 User: Let's do it.
6 Bot: Please tell me your check-in time.
7 User: 1.1.18
8 Bot: Please tell me your check-out time.
9 User: 2.1.18
10 ...
11 Bot: I found 4 items. The first 3 are: 1. Einzelzimmer, 2. Doppelzimmer, 3. Doppelzimmer Superior.
12 User: 2
13 Bot: You can buy an offer of this room.
14 User: Let's do that.
15 ...
16 Bot: Buy action completed. You can see the confirmation.
```

Demo: <https://bot.dialogflow.com/3aa58719-b665-4e7b-970a-564c1b9a64c5>

[Şimşek & Fensel, 2018]

4. Ongoing Work

- Further definition of constraints and extension on schema.org (actions) with SHACL
 - Defining restrictions relevant types (e.g. WebAPI)
 - Extension for different authentication methods
 - Formalizing the relevant semantics of schema.org
- Tools to annotate Web APIs with schema.org actions

4. Ongoing Work

<https://actions.semantify.it/vocab> - For the vocabulary description

Schema.org Web API Description Vocabulary Entites A-Z Classes Properties Shapes Statistics

webapi:WebAPIShape leaf node

```
graph TD; shShape[sh:Shape] --- webapiWebAPIShape[webapi:WebAPIShape]
```

URI

<https://actions.semantify.it/vocab/WebAPIShape>

Label

WebAPIShape

Target Classes (1)

- [schema:WebAPI](#)

Conclusion and Future Work

- Semantic description of Web APIs is crucial for automated consumption by APIs
- schema.org: de-facto industrial standart
 - good indication for adoption
- Semi-automated generation of conversational interfaces
- Next step:
 - finalize the core vocabulary
 - more pilots

References

[Şimşek & Fensel, 2018] Şimşek, U and Fensel, D. (2018). Intent Generation for Goal-Oriented Dialogue Systems based on Schema.org Annotations. First International Workshop on Chatbots co-located with ICWSM 2018, Palo Alto, CA

[Lanthaler & Gütl, 2011] Lanthaler, M., Gutl, C., 2011. A semantic description language for RESTful Data Services to combat Semaphobia, in: 5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011), IEEE, Dajeon, Korea. pp. 47–53.

[Zaveri et. al., 2017] Zaveri, A., Dastgheib, S., Wu, C., Whetzel, T., Verborgh, R., Avillach, P., Korodi, G., Terry, R., Jagodnik, K., Assis, P., Dumontier, M., 2017. smartapi: Towards a more intelligent network of web apis, in: Blomqvist, E., Maynard, D., Gangemi, A., Hoekstra, R., Hitzler, P., Hartig, O. (Eds.), The Semantic Web, Springer International Publishing, Cham. pp. 154–169.

[Verborgh et. al., 2012] Verborgh, R., Steiner, T., Van Deursen, D., Coppens, S., Valls, J.G., Van de Walle, R., 2012. Functional Descriptions As the Bridge Between Hypermedia APIs and the Semantic Web, in: Proceedings of the Third International Workshop on RESTful Design, ACM, New York, NY, USA. pp. 33–40. doi:10.1145/2307819.2307828.

[Lanthaler & Gütl, 2013] Lanthaler, M., Gutl, C., 2013. Hydra: A vocabulary for hypermedia-driven web APIs. CEUR Workshop Proceedings 996.