

Adding Rules on Existing Hypermedia APIs

Michael Petychakis
National Technical University of
Athens
School of Electrical and Computer
Engineering
Athens, Greece
mpetyx@epu.ntua.gr

Fenareti Lampathaki
National Technical University of
Athens
School of Electrical and Computer
Engineering
Athens, Greece
flamp@epu.ntua.gr

Dimitrios Askounis
National Technical University of
Athens
School of Electrical and Computer
Engineering
Athens, Greece
askous@epu.ntua.gr

ABSTRACT

During the past years, the data deluge that prevails in the World Wide Web has been accompanied by a number of APIs that expose business logic. In this paper, we discuss a novel approach to enrich existing API standards definitions with business rules. Taking advantage of the REST principles, we aim at enabling the creation of generic clients that can dynamically navigate through semantically enriched web affordances with the help of Hydra-based Hypermedia API descriptions, which encapsulate the finite state machine of possible actions into SWRL rules.

Categories and Subject Descriptors

H.5.4 [INFORMATION INTERFACES AND PRESENTATION]: Hypermedia – *Architectures, Navigation, Theory.*

General Terms

Documentation, Design, Standardization

Keywords

Web APIs, Hypermedia, Affordances, Semantic Web, Intelligent Agents, Web Services

1. INTRODUCTION

The ever-accelerating growth of web services and the prevalence of multi-sided business models have created an unprecedented number of custom and often non-standardized services, yet often hinder seamless development and easy maintenance of web and mobile applications. The affordances over web objects, used by such services, are also limited to certain applications that developers design while their lifespan is the same or even smaller than the specific application lifecycle.

Since its very beginning, the World Wide Web has aimed at making information sharing easier for people all over the world. A great outbreak of technologies and practices has accordingly emerged, leading to the proposal of several architectures for web services and software in general over the years. REST, which was

proposed by Roy Fielding on his dissertation [1], is listed among the most popular and influential propositions, that mainly due to its simplicity, has inspired a vast amount of different applications over the last decade. However, despite its prevalence over time, the problem of sharing information and conducting seamless transactions still persists and, with the rise of the Internet of Things, demands new generic solutions to be delivered.

In this context, the underlying technological state-of play is still not adequate for the creation of artificial agents that could autonomously execute tasks and responsibly and dynamically decide about their instructed goals over the web. The present paper aims at briefly discussing the current situation regarding the core web technologies, including REST, APIs and Hypermedia APIs, and outlining their connection to the Semantic Web technologies. It also highlights the importance of business rules over the web by analyzing the current situation. The core proposition of this paper is the DeepGraphs specification for designing APIs, in a way that renders reasoning over the client-server transactions more efficient and safer. The paper concludes that extending API descriptions with rules could probably provide sufficient mechanisms to create both generic clients and servers that would eventually lead to simplifying reasoning over the web of services and transactions, and proactively providing necessary mechanisms to be put into use in the imminent Internet of Things.

2. ADDING RULES ON TOP OF HYPERMEDIA APIS

2.1 Rules on the Web

The web in its current format has been extended mainly by the needs of people and the companies in a demand-driven way, rather than adhering to a specific plan. The various constraints coming along with the data produced can be attributed to this spontaneous growth.

A lot of work has actually already been undertaken by W3C (WWW Consortium) to describe the various data constraints with the OWL standardization group and later with the SWRL and Rule Interchange Format (RIF) working groups. Such groups have also succeeded in addressing interoperability aspects between the various rule languages, according to the needs of enterprises as well as the academia. SWRL (Semantic Web Rule Language) is a popular rule definition language, accompanied by supporting software implementations (e.g. Protégé). Despite having a lot of dialects, RIF has also achieved to provide a representation of all important rule languages. Its main motivation has been the existence of several languages (SWRL, RuleML, R2RML, Datalog, F-logic) and the more and more pressing need to make all systems depending in such languages communicate with each other. Actually, RIF has not yet received the adoption it was

Copyright is held by the International World Wide Web Conference Committee (IW3C2). IW3C2 reserves the right to provide a hyperlink to the author's site if the Material is used in electronic media. *WWW'15 Companion*, May 18–22, 2015, Florence, Italy. ACM 978-1-4503-3473-0/15/05. <http://dx.doi.org/10.1145/2740908.2743041>

expected, but its prospects are very promising with the rise of the Web of Things.

2.2 Hypermedia APIs

The web, as experienced today, is an elaborate world of links, with people following subsequent links to reach content and constantly beginning new journeys from link to link. Nowadays, a new concept, often known as hypermedia, has emerged from the need to share links through the APIs, in order to allow agents to automatically browse through URLs and find the resource they actually need within the provided API.

An informal descriptive definition has been provided by Amundsen [2], mentioning that “Hypermedia is a way for the server to tell the client what HTTP requests the client might want to make in the future. It’s a menu, provided by the server, from which the client is free to choose. The server knows what might happen, but the client decides what actually happens”.

Currently, a lot of work is undertaken to formalize and standardize hypermedia descriptions in a similar manner like HTML. From the moment HTML was formalized, it became easier to create generic web browsers that rendered almost any web page available in the Web. By adopting a similar philosophy, generic agents which handle a variety of APIs from different resources and execute their specific goals, could arise. A noteworthy approach has been proposed by Lanthaler [3] on Hydra, defining a lightweight Linked Data vocabulary that aims to describe the semantics of creating Hypermedia web services on top of Semantic Web technologies.

2.3 DeepGraphs Specification

The main concept behind DeepGraphs is that a server which complies with a Hypermedia standard can actually “help” any client to explore the way it can accomplish its desired goal, by providing guidance on the map of its entry points and the logic behind the implementation and its semantics. What we propose is essentially an extended Hydra-based Hypermedia API description in the form of a JSON-LD file, which encapsulates the finite state machine of possible actions into SWRL rules. By reusing common vocabularies based on RDFs and OWL, generic clients that “know” how to accomplish a specific goal shall be created.

In summary, the client which has a goal to achieve receives a DeepGraphs document, creates the transaction execution map by analyzing the API atlas (as specified in the DeepGraphs document) and starts running each state sequentially, according to rules. Following the REST paradigm and since GET requests can be easily cached, the server responds to each request independently.

An example that demonstrates in practice the main principles of the DeepGraphs methodology is described in detail in the following lines: The assumption is that an end-user of a third-party application wants to buy a copy of the Ulysses book from an online store that follows the DeepGraphs paradigm and the schema.org conventions of terminology.

The generic client that the application uses only knows that it needs to accomplish an action, namely `schema.org/BuyAction` (Buy), concerning a specific resource, in particular `schema.org/Book` (the book) with property name “Ulysses”. Since it does not know how the server actually “works”, it sends a “GET” request on the base path (which, for convenience purposes, is commonly declared as the Entry path in the Hypermedia terminology). The server responds with a valid DeepGraphs document (in its API response) that contains the

hypermedia map of the URLs it contains, and their descriptions so that the application client has at its disposal the relevant “API atlas” of the bookstore server.

The necessary starting point and the exact workflow of the steps to be executed are provided within the DeepGraphs document through SWRL rules that represent a series of steps, which a client can follow. The bookstore server can be thus viewed as a FSM (Finite State Machine) that actually implements any workflow for the client according to the SWRL ruleset. It needs to be noted that SWRL was chosen as a W3C standard that combines the logic of rules (if-this-then-that) with the existing OWL stack.

When a state in the overall transaction becomes “True”, the client “discovers” the next state to follow through the hypermedia map of the API. Subsequently, it sends the next request to the server which is responsible for reasoning over this request. This is a repetitive situation for the overall transaction, until the client reaches the final goal or the server starts denying to handle the client’s requests.

In general, the server can deny a transaction if an Integrity Constraint (IC) is violated or whenever a specific rule of Denial (False) is triggered. For every request, the IC of the transaction is checked, although it is acknowledged that such requests may have an exponential impact of complexity per transaction, according to the work of Sirin [4] and Cali [5].

According to the proposed DeepGraphs methodology, the Resources and the Actions are described based on Linked Data vocabularies and the Semantic Web standards, permitting the server and the client to be implementation-agnostic and unaware of predefined sets of actions. Therefore, new affordances can be added dynamically to existing resources and their behavior can be determined according to existing or new rules. For example, if a new affordance like “Search” Action is added and extends the existing “General” Action over a list, then the client (or agent) has at least a starting point, which is the “General” Action.

This dynamic resolution of new affordances by the server is not yet deeply explored in the literature. There are few approaches, such as indicatively by Web Intents or Verborgh **Error! Reference source not found.**, but not much of those have received any major adoption yet.

3. GENERIC IMPLEMENTATIONS FOR PROVIDER AND CONSUMERS

The World Wide Web harnesses the power of distributed intelligence amongst people and machines, publishing information at tremendous velocity and volume. In order to retrieve such information across the globe, search engines come to the foreground. With regard to APIs, a preliminary approach for discovering APIs is the API directory, ProgrammableWeb. However, if APIs were exposed in a format in accordance with the DeepGraphs specification, then a new paradigm of search engines would be created for APIs.

The capabilities that have made the web useful for people can also work for artificial intelligent software agents. Such agents actually already exist in the form of any automated client and have also contributed to the current state-of-play of the web. However, the kind of agent described in this paper diverts from existing agents since it is envisioned to dynamically interact with new objects according to their needs.

In this context, DeepGraphs aims to become a specification for the Internet of Things, acting as an interoperability layer between server and client implementations that anticipates the behavior of

the ever-growing sensors in the Web. Without the potential for reasoning, APIs and sensors would be just another means of providing data, not dynamically discovering hidden affordances and knowledge, which is the target of this paper.

3.1 Designing a new API

While designing a new DeepGraphs API, the developer in principle needs to define a finite state machine of all the possible situations of the client-server interaction in SWRL. To describe the API, any Hypermedia standard may work to show the map of all the possible paths that the client has to follow.

To increase adoption by the API designers, we currently study the mappings of the Hydra specification to other popular standard, such as swagger, RAML and API-Blueprints in order to provide an easy way to integrate existing expertise and technological infrastructures.

3.2 Designing a new client

Clients in DeepGraphs are differentiated to what has already been widely adopted in the web nowadays. Clients are viewed as generic reasoners that, depending on their goals, follow a specific path according to the event-driven reasoning that has been sent to them by the server. In practice, they may consume the entire map by the Hypermedia API through the DeepGraphs document, while having in parallel at their disposal the rules to guide them on what to do next at every moment. Following this rationale, a DeepGraphs client could interact with multiple servers simultaneously or even implement combined transactions on those servers.

4. CONCLUSION AND FUTURE WORK

A paradigm shift in enterprise-centric design of business applications is emerging based on the proliferation of APIs that play a pivotal role in a thriving API ecosystem by unlocking latent value in data and information of available services. With the purpose of supporting easy integration of a broad spectrum of existing or new functionality in a platform-independent way, the DeepGraphs specification was initiated and creates a novel, generic way for services to expose their functionality and for clients to dynamically comprehend such functionality.

The impact of a technology like DeepGraphs on wearable devices and smart houses is invaluable as it may act as a bridge between the current approaches of the web and the semantic web. In practice, this migration would be effortless since developers would require no deep understanding of the mathematical, academic background and they could continue to use the same tools. The proposal of the DeepGraphs specification aims to be a technology, simple to understand and adopt by the developers.

In this direction, we have already created a UI mechanism for developers to easily document their existing APIs in the OPENi API Builder [7] and we plan to extend it according to this proposal in order to facilitate the DeepGraphs adoption.

Privacy is among the key aspects in our next steps. There are already many discussions on W3C about privacy, and more recently about a Payment API. A key concern on those discussions that also affects the DeepGraphs approach is how it is

ensured that the server is not going to send any malicious software.

Another important aspect we currently explore, is how our methodology could be extended to include more Hypermedia specifications in a more generic way and how to represent SWRL rules into JSON-LD in the most effective way. Since SWRL is partially interoperable with RuleML, RIF, R2ML and other popular rule languages, we plan to extend DeepGraphs to support more specifications on describing rules. We currently base our approach on the mapping provided by Ma [8] so that we can have a basis to support the various formats.

Finally, since the DeepGraphs methodology is heavily based on reasoning, we experiment with the available reasoning approaches with a view to extend the DeepGraphs specification towards an optimal algorithmic implementation of the various potential states of affordances in the API atlas.

5. ACKNOWLEDGMENTS

This work has been created in the context of the EU-funded project OPENi (Open-Source, Web-Based, Framework for Integrating Applications with Social Media Services and Personal Cloudlets), Contract No: FP7-ICT-317883.

6. REFERENCES

- [1] Fielding, Roy Thomas. "Architectural styles and the design of network-based software architectures." 2000.
- [2] Richardson, Leonard, Mike Amundsen, and Sam Ruby. RESTful Web APIs. "O'Reilly Media, Inc.", 2013.
- [3] Lanthaler, Markus. "Leveraging Linked Data to Build Hypermedia-Driven Web APIs." REST: Advanced Research Topics and Practical Applications. Springer New York, 2014. 107-123.].
- [4] Sirin, Evren, Michael Smith, and Evan Wallace. "Opening, Closing Worlds-On Integrity Constraints." OWLED. 2008.
- [5] Cali, Andrea, Georg Gottlob, and Thomas Lukasiewicz. "Datalog±: a unified approach to ontologies and integrity constraints." Proceedings of the 12th International Conference on Database Theory. ACM, 2009.
- [6] Verborgh, Ruben, et al. "Distributed affordance: an open-world assumption for hypermedia." Proceedings of the 22nd international conference on World Wide Web companion. International World Wide Web Conferences Steering Committee, 2013.
- [7] Petychakis, Michael, et al. "Enterprise Collaboration Framework for Managing, Advancing and Unifying the Functionality of Multiple Cloud-Based Services with the Help of a Graph API." Collaborative Systems for Smart Networked Environments. Springer Berlin Heidelberg, 2014. 153-160.
- [8] Ma, Z. M., and Xing Wang. "Rule Interchange in the semantic web." Journal of Information Science and Engineering 28.2 (2012): 393-4.