RESTful Conversation with RESTalk

The Use Case of Doodle

Ana Ivanchikj(✉)

Faculty of Informatics, University of Lugano (USI), Lugano, Switzerland
ana.ivanchikj@usi.ch

Abstract. With the availability of multiple Web services, offering identical or similar utilities, their ease of use has become a valuable success factor, highly influenced by API's documentation quality. Tools are available for documenting the various technical details pertaining to the static structure of RESTful services. Additionally, we have identified interest in and usefulness of also depicting API's behaviour, i.e., the viable RESTful conversations defined as multiple client-server interactions necessary to utilize certain service functionality. RESTalk, the REST domain specific language we have designed for modeling RESTful conversations, facilitates the conceptual modeling and visualisation of API's behaviour. In this poster paper, we extend RESTalk with new language constructs and apply it on a real RESTful API, the Doodle API, which refers to RESTful conversations between multiple clients and one server.

Keywords: RESTful web services · Multiparty conversations · Domain Specific Language · Real API · Use case · Doodle

1 Introduction

The idea behind the Representational State Transfer (REST) architectural style is to introduce simplicity through constraints and standardization. This has inevitably brought about its popularity, with an ever-growing number of Web Application Program Interfaces (APIs) declaring REST compliance. While the awareness for the need of documenting the low-level HTTP details of REST API's static structure has resulted with several tools, such as RAML or Swagger[1], there is still a lack of support for conceptual modeling and visualisation of REST API's dynamics. Reaching a certain resource state, frequently requires undertaking a predefined sequence of interactions or choosing among different alternative paths, thus shifting from the concept of a single RESTful interaction to the concept of a RESTful conversation [1]. As conversations become more complex, visualising them can help decrease the cognitive load of API designers, who need to communicate their design to the developers, or of API clients, who need to use the API. Currently, mainly UML Sequence Diagrams or UML State Machines are used to model API's behaviour. However, they lack the ability

[1] http://raml.org; http://swagger.io.

A. Bozzon et al. (Eds.): ICWE 2016, LNCS 9671, pp. 583–587, 2016.
DOI: 10.1007/978-3-319-38791-8_53

of concisely presenting important REST tenets, such as HTTP methods, URIs, headers, hypermedia flow, status codes, etc.

Therefore, in [2] we have proposed the first version of RESTalk, a Domain Specific Language (DSL) for modeling RESTful conversations. In [3] we have conducted an exploratory survey, among practitioners and researchers, to obtain initial feedback on its cognitive characteristics, and its perceived usefulness. In this poster paper, based on the survey feedback, we extend RESTalk with additional language constructs to support modelling of multiple participants in the conversation and their roles, as well as to present resource state transitions, and asynchronous email communication. We use this extended version of RESTalk to model the behaviour of Doodle, a well known scheduling Web service[2] which depicts RESTful conversations between multiple clients and one server.

2 RESTalk

To model RESTful interactions, Nikaj and Weske [4] add REST-specific annotations to the Choreography diagram of the Business Process Model and Notation (BPMN) 2.0 standard. With RESTalk we go further and modify the standard notation in order to visually draw the attention to what is essential in RESTful interactions, i.e., the content of the exchanged messages.

The results from the above mentioned exploratory survey [3], have revealed missing constructs in RESTalk for depicting: 1) roles of the conversation participants, when viable interactions depend on such roles; 2) multiparty conversations where multiple clients interact with the server; 3) client-server interactions which lead to changes in the related resource's state; and 4) asynchronous interaction through email notifications sent to participants. Thus, in this poster paper, we propose a way to extend RESTalk to represent such important details. Figure 1 shows a summary of all the language constructs relevant to the Doodle use case which we will describe in Sect. 3. The semantics of the existing RESTalk constructs, i.e., the ones not added to the language with this poster paper, are explained in detail in [2].

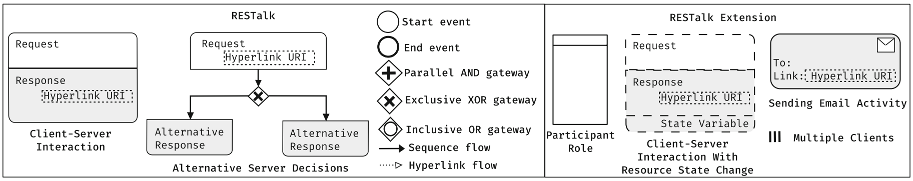

Fig. 1. RESTalk Constructs

[2] http://doodle.com.

3 The Doodle Use Case

To show-case the appropriateness of RESTalk for documenting real RESTful APIs, we have decided to use it to model Doodle's API. Doodle's basic free offering is a Web service which facilitates event scheduling between multiple participants. We have opted for modeling this basic service, where no log-in is required. Due to the lack of detailed API documentation[3], we have used a prescriptive instead of descriptive approach in modeling its dynamics.

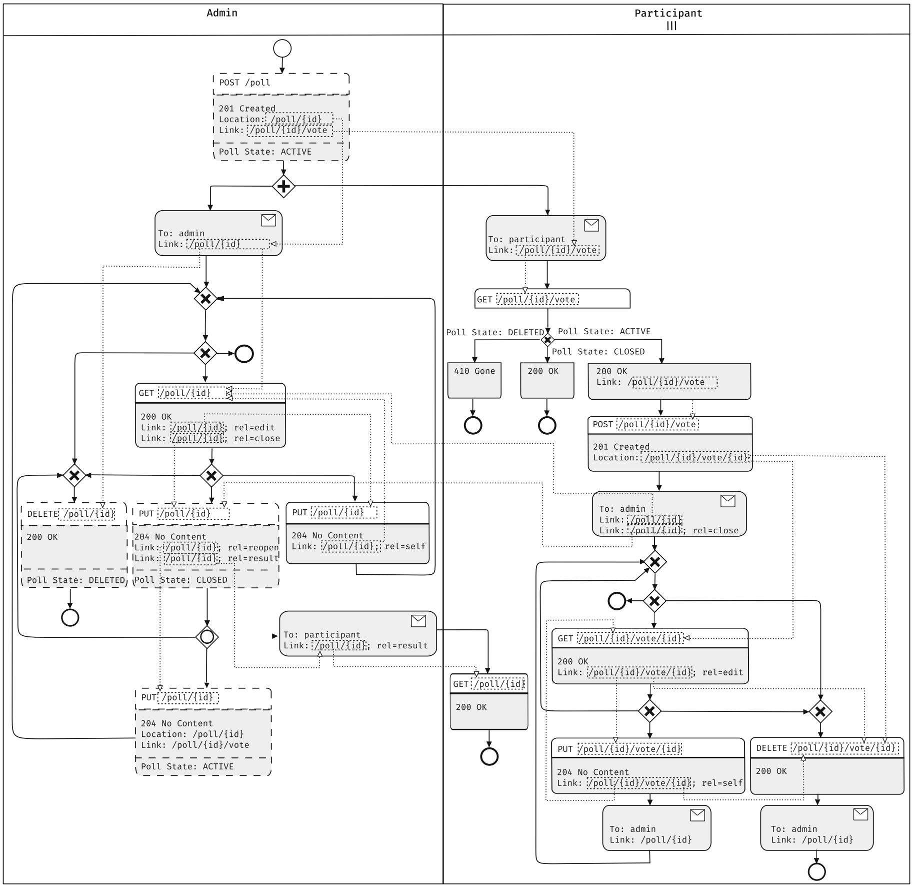

Fig. 2. Doodle RESTful conversation with RESTalk

Following is the natural language description of the conversation modelled in Fig. 2. The client can take two distinct roles leading to different privileges. The admin role, assigned to the client who has created the poll, and the participant

[3] http://support.doodle.com/customer/en/portal/articles/664212.

role, assigned to the clients who are invited to participate in the poll. After the admin creates the poll, emails are sent to the admin, with a link to the poll resource, and to all participants, with a link to the poll voting resource. The poll can have three states: active, closed, or deleted. The interaction which results with entering in a certain state is marked with a dashed line and the state variable is noted under the response box. The admin can edit, close or delete the poll at any time, as well as reactivate it if closed. When the poll is closed, an email notification may be sent to the participants. The participants can only vote, change or delete their votes when the poll is active, resulting with an email being sent to the admin. If the poll is closed, the participants can only see the results, without editing them. If the poll is deleted, they get an error message.

4 Related Work

Although the existing work on modeling RESTful APIs distinguishes between the need of modeling both the static and dynamic aspects of these APIs [5], the focus has mainly remained on the static aspect. When tackled, REST API's dynamics has been mostly analysed from the resource state transition perspective [6,7]. We add another perspective to the dynamics, i.e., the client-server interactions necessary to initiate such state transitions. To visualise them, we have designed and work on improving RESTalk, a DSL for modeling RESTful conversations.

5 Conclusions and Future Work

It is rarely possible to follow a waterfall approach when designing a DSL, as evidenced by the continuous release of new versions of many standard languages. Each DSL keeps on evolving as its domain is evolving, and as users provide feedback on identified gaps in the language, or situations which can not be reliably modeled with it. RESTalk is not an exception to this practice, and we support the agile approach in language design with frequent iterations. Thus, in this poster paper, we extend RESTalk based on recently acquired survey feedback and showcase the appropriateness of using such extended version for modeling the Doodle API. Further extensions are planned to develop a version complete enough to be launched for testing in industry. In the future the notation could be supported with a multi-view editor and a tool for automatic code generation.

References

1. Haupt, F., Leymann, F., Pautasso, C.: A conversation based approach for modeling REST APIs. In: Proceedings of the 12th WICSA 2015, Montreal, Canada, pp. 165–174 (2015)
2. Pautasso, C., Ivanchikj, A., Schreier, S.: Modeling RESTful conversations with extended BPMN choreography diagrams. In: Weyns, D., et al. (eds.) ECSA 2015. LNCS, vol. 9278, pp. 87–94. Springer, Heidelberg (2015). doi:10.1007/978-3-319-23727-5_7

3. Ivanchikj, A., Pautasso, C., Schreier, S.: Visual modeling of RESTful conversations with RESTalk: an exploratory survey. J. Softw. Syst. Model. (2016, Under review)
4. Nikaj, A., Weske, M.: Formal specification of RESTful choreography properties. In: Proceedings of the ICWE 2016, Springer (2016) (to Appear)
5. Schreier, S.: Modeling RESTful applications. In: Proceedings of the Second International Workshop on RESTful Design, pp. 15–21. ACM (2011)
6. Rauf, I.: Design and validation of stateful composite RESTful web services. Ph.D.thesis, Turku Centre for Computer Science (2014)
7. Mitra, R.: Rapido: A sketching tool for web API designers. In: Proceedings of the 24th WWW 2015, Florence, Italy, pp. 1509–1514 (2015)