A-Match: Facilitating Data Exchange Between Different Applications via API Matching

Sarah Böning, Max Möbius, Katharina Juliane Popendicker

To answer challenging research questions, there are often several distinct tools and software necessary. Their successive execution is also called a toolchain. Toolchains often need interchangeable data between their tools. Especially in the engineering context, there is a heterogeneous software landscape and the different tools often have different data formats and conventions.

Formatting the data output from one tool to be correctly loaded by a second is a tedious manual process. Some tools offer different export formats, which can reduce the effort to some extent. However, this does not guarantee an error-free import and manual checks are needed as well. Additionally, compatibility can be withdrawn by software updates. This makes a complete automatic toolchain nearly impossible. A more reliable way to ensure correct data exchange is by interconnecting compatible interfaces (APIs) of the tools.

The idea of matching two APIs is not new. There are several approaches in research [1-5]. However, these methods often focus on constricted use cases and have different results on different domains. Similar to A-Match, many often rely on ontologies to integrate domain knowledge into the matching system.

As a first step in the direction of automatic API matching, we have developed a prototypically user interface (UI) to offer a fast and correct manual match between data of two different tools in the space domain. With this project, we wanted to extend the functionality of our prototype and adapt it to other contexts, namely the engineering domain. The resulting tool is called A-Match.

A-Match consists of two parts: the user interface (UI) and the matching backend. The user can select the APIs and data objects to match. Then, they can combine the terms as needed by hand or get automatic matching suggestions calculated by the backend. Here, a combination of semantic distance metrics and ontologies defining synonyms are used. When the user is content with the matched terms, the resulting changes are directly sent to the second API as an update. If individual attributes contain units of measurement, it is ensured that the input value is converted to the output unit. Thus, the toolchain can correctly continue.

To ensure that the NFDI4Ing community's needs and wishes were incorporated throughout the project, we held two workshops. The given feedback was then implemented into our product. In the first workshop, we assessed the state of the current implementation and gathered more functionality requirements and wishes. After their implementation, we held a second workshop. Here, we focused on usability to ensure we are delivering a well-designed tool that has the support of the NFDI4Ing community. Additionally, we conducted a user study to evaluate the usability, usage frequency, and usefulness of A-Match. We made our resulting, finished software available as Open Source at the end of the project. A-Match could still be extended to more domains and thus, add more functionality in following projects. It would also be possible to integrate and adapt A-Match into a workflow of a company in an industrial cooperation.

In this presentation, we report A-Match's functionality and results from the project. At the end, we will give a live demo of *A-Match*.

Literature

[1] Wu, Chen; Dillon, Tharam; Chang, Elizabeth. Intelligent matching for public internet web services towards semi-automatic internet services mashup. In: 2009 IEEE International Conference on Web Services. IEEE, 2009. S. 759-766

[2] Khorasgani, R. R., Stroulia, E., & Zaiane, O. R. (2011, September). Web service matching for RESTful web services. In 2011 13th IEEE International Symposium on Web Systems Evolution (WSE) (pp. 115-124). IEEE.

[3] Caragea, D., & Syeda-Mahmood, T. (2004, May). Semantic api matching for automatic service composition. In Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters (pp. 436-437).

[4] Xie, W., Peng, X., Liu, M., Treude, C., Xing, Z., Zhang, X., & Zhao, W. (2020, November). API method recommendation via explicit matching of functionality verb phrases. In Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 1015-1026).

[5] Legler, F., & Naumann, F. (2007). A classification of schema mappings and analysis of mapping tools. Datenbanksysteme in Business, Technologie und Web (BTW 2007)–12. Fachtagung des GI Fachbereichs" Datenbanken und Informationssysteme"(DBIS).