Final Report

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

General Information

Applicants	Diana Peters, Sarah Böning
Institution	Institut für Datenwissenschaften, Deutsches Zentrum für Luft- und Raumfahrt e.V.
Project Title	Erleichterung von Datenaustausch zwischen verschiedenen Anwendungen mittels API-Matching – A-Match
Funding Period	01.07.2021 – 30.06.2022
List of Publications	• A-Match Source Code, Zenodo • NFDI ToolTalk: A-Match - Facilitating Data Exchange between different Applications via API Matching, YouTube

Executive Summary

Digitalization has changed many aspects of our lives and it also transformed research. There are several software tools necessary to answer research questions. For example, calculations, statistics, visualizations, and modelling are all done with software. Especially in engineering science, there is a vast heterogenous software landscape. This hinders easy interoperability between tools. However, this is much-needed for the so-called toolchains. A toolchain is the sequential execution of different software tools to answer a specific question. Ideally, this happens automatically. However, because of the heterogenous software landscape and the resulting different data formats and conventions across programs, data conversions between toolchain steps are needed. Doing this by hand is not feasible as it is very time-consuming the more tools are used. A more reliable way to ensure correct data exchange is by interconnecting compatible interfaces (APIs) of the tools.

The goal of the *A-Match* project was to build a system that can match attributes of two different APIs to facilitate data exchange for toolchains. For that purpose, we extended an existing prototype developed in-house at the Institute of Data Science of the German Aerospace Center (DLR) and adapted it to the needs of the NDFI4Ing community.

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 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 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.

Work- and Result Report

Project Goal

To answer challenging research questions, there are often several distinct tools and software necessary. This so-called toolchain often needs interchangeable data between the 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.

As a first step in this direction, we have developed a prototypical user interface (UI) to offer a faster 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.
At the end of the project, the source code for our tool, *A-Match*, was released Open Source to make it available to the engineering community.

Development

We defined three work packages and milestones for this project. In the following, we describe the development by reference to the packages.

Work Package 1: Workshop and Definition of Requirements

The first workshop at the beginning served to refine the implementation scope of the project. Based on the feedback, we derived the needed features of the user interface.
We planned to achieve this milestone after the first month. However, the workshop preparation required more time than anticipated and we held the workshop and formulated the requirements at the end of month 2.

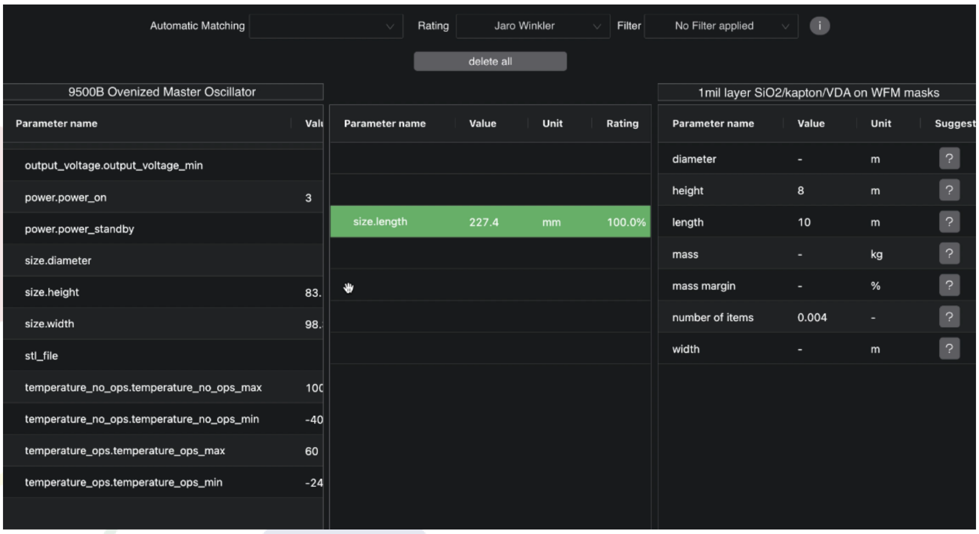
Work Package 2: Functional Requirements

After month 8, we planned to have every functional requirement formulated in WP 2 implemented in the *A-Match* user interface (milestone 2). We focused on adding functionality rather than usability. However, normal usage should be possible.
We achieved this milestone within our timeframe. Please refer to the section *Results* for more details.

Work Package 3: User Study and UI Refinements
After implementing the requested features, we wanted to optimize the usability of *A-Match*. Thus, we held a second workshop to retrieve feedback regarding the design of the interface. We conducted a user study to evaluate the usability of *A-Match* and derived requirements for UI refinements. These were then implemented.
This was the third and final milestone. It was achieved within the time frame at the end of month 12.

Results

As mentioned in the previous section, we reached all three milestones. We have developed a UI that facilitates data exchange between two software tool that offer an API. To include APIs in A-Match, an adapter has to be implemented for each. Then, on the first page of the UI, the user can select the two APIs to match. Here, we distinguish between input- and output API. This is important, as the input terms will be matched to the output terms and the result is sent to the output API for updates.
Next, the user can select the input and output data structure, or object, to match. The *A-Match* UI offers several filter options to find the right objects. On the Matching Page, the different attributes from the objects are matched. Here, the user can match manually by dragging and dropping or let *A-Match* compute the best matches automatically. This is done via a request to the *A-Match* backend. To compute the best match, several semantic distances are used as well as a synonym ontology. Users can use the default ontology or upload their own. *A-Match* uses visual clues and ratings to indicate the match quality. Figure 1 shows the matching page.
Additionally, we added info buttons with explanations to every page of the *A-Match* UI for first time users.



Automatic Matching
Rating: Jaro Winkler
Filter: No Filter applied

9500B Ovenized Master Oscillator

Parameter name	Valı
output_voltage.output_voltage_min	
power.power_on	3
power.power_standby	
size.diameter	
size.height	83.
size.width	98.
stl_file	
temperature_no_ops.temperature_no_ops_max	10(
temperature_no_ops.temperature_no_ops_min	-40
temperature_ops.temperature_ops_max	60
temperature_ops.temperature_ops_min	-24

Parameter name	Value	Unit	Rating
size.length	227.4	mm	100.0%

1mil layer SiO2/kapton/VDA on WFM masks

Parameter name	Value	Unit	Suggest
diameter	-	m	?
height	8	m	?
length	10	m	?
mass	-	kg	?
mass margin	-	%	?
number of items	0.004	-	?
width	-	m	?

Figure 1: Screenshot of the Matching Page

Lastly, the user can save the matching. Here, the UI summarizes the matched terms and highlights possible incompatible matches that will not be saved. The user can now change their matches or proceed. The updated data is sent to the output API and can be exported and downloaded in different formats (JSON, XML, CSV) if needed. Terms matched by the users update the synonym ontology and thus, improves the automatic matching later. Now, more objects can be matched as needed, or the UI can be closed.

Tables 1 and 2 list all implemented features of *A-Match* within the scope of this project. During the course of the project, we considered feedback from the NFD4Ing community and focused on functionality and usability alike.

Area	**Requirement**	**Importance**
Functionality	F1 The matching is saved offline in a file.	High
	F2 The software supports versioning and logging to trace the matches.	High
	F3 The UI supports the upload of different ontologies (either via file upload or URI link).	High
Website	W1 There is a documentation on how to use the matching software.	High
	W1.1 The repository contains a general documentation.	
	W1.2 The repository contains a pictured and illustrated example with Part DB and VirSat.	
Other	O1 The website offers a reasoning for the automated match	Medium

Table 1: Requirements from the first workshop

Area	**Requirement**	**Importance**
Start Page	St1 There is an info button referencing the documentation on how to add your own APIs.	High
Object-Selection Page	OS1 A pop-up with all properties of the selected object is shown when clicking it/hovering on it.	High
	OS2 There is a notification when a filter option is selected that is invalid or not available.	High
Object-Matching Page	OM1 Above the matching table, there is a text to explain the "useful" rating rank for each selected metric (& and a note to the info button).	Medium
Saving Page	Sa1 The generated file for exporting contains a header detailing the 2 APIs & objects that were matched.	High
Other	O1 When the website is opened for the first time by a user, there is a pop-up notification to make them aware of our info buttons.	Medium

Table 2: Requirements from the second workshop

Discussion

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.

We developed and extended *A-Match* with the engineering community in this project. Thus, we think that it can be successfully applied there.

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.

Cooperation & Exchange

The project had a tight cooperation and exchange with NFDI4Ing. We held two workshops and invited participants from the NFDI and NFDI4Ing community mail lists. These workshops gave direction to the development of *A-Match*. We derived requirements from the participants' feedback in the first workshop. These were then all implemented. In the second workshop, we gathered usability feedback for final improvements that we then implemented until the end of the project.

Additionally, we showed *A-Match* at an NFDI Tool Talk and plan to present the outcome of the project at the NFDI4Ing conference 2022.

Relevancy within the Community

We directly aim for users from the NFDI4Ing community with *A-Match* and integrated its members during the whole course of the project. Thus, we hope that it has a high relevancy to the community.

This is also reflected in the survey we conducted in the second workshop. Here, participants were asked to rate the usefulness of A-Match in general, for their field, and for their work. On average, the usefulness to the respecting field was rated 3 out of 5. For more details, see Figure 2. As the participants were all invited from NFDI and NFDI4Ing mailing lists, we can derive that A-Match is relevant to the NFDI4Ing community.

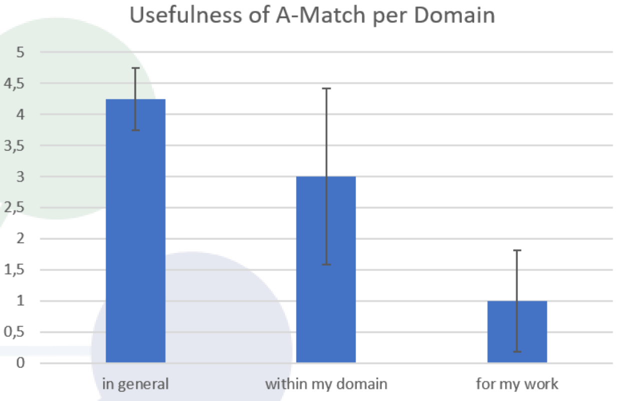


Figure 2: Results of the perceived usefulness

Usage within the Community

To ensure the usefulness of *A-Match*, we included the NFDI4Ing community. Requirements and wishes were gathered in two workshops and then implemented. Additionally, we asked participants to rate their perceived usefulness of *A-Match* during the user study. Every participant expressed interest to use *A-Match*. The answers regarding the frequency varied, but half of the participants answered they would use it at least once or several times per year.

Publications

During and after the project phase, we tried to make *A-Match* visible. Firstly, we invited participants to the workshops via NFDI and NFDI4Ing mailings lists, but also made the invitations public on the NFDI4Ing website [6, 7]. We also presented *A-Match* as a Tool Talk of the NFDI directorate [8].

Additionally, information and news regarding *A-Match* were published on the social media accounts [9, 10] and website of our institute [11].

At the end of the project, we published the source code on Zenodo [12], a well-known Open Source platform. Finally, we plan to present *A-Match* and the results at the NFD4Ing Conference [13] 2022.

Acknowledgements

We would like to thank the Federal Government and the Heads of Government of the Länder, as well as the Joint Science Conference (GWK), for their funding and support within the framework of the NFDI4Ing consortium. This project was partially funded by the German Research Foundation (DFG) - project number 442146713.

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).

[6] https://nfdi4ing.de/wp-content/uploads/2021/07/Einladung-Workshop-A-Match.pdf, accessed 28.06.2022

[7] https://nfdi4ing.de/wp-content/uploads/2021/12/Einladung-Workshop-2-A-Match.pdf, accessed 28.06.2022

[8] https://www.youtube.com/watch?v=Y0riA4wux6E, accessed 28.06.2022

[9] https://twitter.com/DLRdatascience/status/1521394199874371584, accessed 28.06.2022

[10] https://twitter.com/DLRdatascience/status/1547086550227369988?s=20&t=AAtkq0sIz_0C_ubRPChBGw, accessed 14.07.2022

[11] https://www.dlr.de/dw/desktopdefault.aspx/tabid-18221/28963_read-76243/, accessed 28.08.2022

[12] Max Möbius, Katharina Juliane Popendicker, & Sarah Böning. (2022). A-Match (v1.0.0). Zenodo. https://doi.org/10.5281/zenodo.6641652

[13] https://nfdi4ing.de/conference, accessed 28.06.2022