EasyChair Preprint

№ 896

Standard APIs and link prediction for the digital thread

Axel Reichwein, Guillermo Jenaro-Radaban and Zsolt Lattmann

April 13, 2019

Standard APIs and Link Prediction for the Digital Thread

Axel Reichwein, Koneksys
Guillermo Jenaro-Radaban, A^3 by Airbus LLC
Zsolt Lattmann, A^3 by Airbus LLC

Digital thread is viewed as a game changer by the US Air Force [DigitalThreadUSAF2013] to increase product development speed and reduce risk. The digital thread is sometimes called digital continuity, or even simply traceability. From a technical perspective, the digital thread is very simply about connecting data across the product life cycle. Even though this may seem very simple, achieving the digital thread is currently a challenge. PLM vendors may claim that they are offering digital thread solutions, but existing solutions have incomplete coverage of product data and do not scale. As PLM vendors are traditionally hesitant to support standards, which are necessary to connect data from 500+ data sources, and as PLM vendors prefer to use proprietary integration approaches in order to achieve vendor lock-in, the coverage of existing digital thread solutions from PLM vendors is limited to CAD-related data and is improving very slowly.

Simultaneously, product manufacturers are trying to get the most value from IoT data as it offers many new business opportunities. IoT data on its own can be analyzed to better understand how products are actually being used in the field. However, IoT can also be used to improve the physics-based models which were used during design and manufacturing [GE-AI-Future-2016]. For example, models predicting the end of life of a product can gain accuracy based on IoT data. As a result, products could potentially be operated longer and their value could significantly increase. In order to improve a specific physics-based model based on some specific IoT data, it is necessary to keep track of what IoT data relates to what physics-based model. Both data sources need to be connected. In other words, the digital thread is necessary for getting the most value from IoT data. As PLM vendors are innovating very slowly to support the digital thread, some organizations are looking at open standards and architectures inspired by the World Wide Web to connect data, and achieve the digital thread independently of PLM vendors, in order to innovate at their own speed. In this context, this white paper will explain the value of standard APIs and link prediction for the digital thread.

Systems engineers, who need to understand the big picture, are also interested in connecting various engineering data. Instead of calling it the digital thread, they call it traceability. Traditional traceability activities in systems engineering are typically performed at the end of a

design activity, to document how requirements relate to tests, and how these tests were performed.Traditional traceability activities involve describing the trace links in a single tool, or in spreadsheets, or in a PLM system. These links are then traditionally inspected during certification activities or when the cause of an accident or malfunction needs to be found.

Modern traceability activities in contrast are meant to be used in a more active way [Future-of-traceability-2016]. The trace links are meant to used during the design activity, in addition to using the trace links after the design activity. The trace links are used to help engineers gain a better understanding of the design in order to take better design decisions. The trace links are meant to be frequently created, modified, navigated, in addition to being inspected. As this modern traceability activity becomes more common, engineers will more easily and more precisely understand what they can reuse across projects.

The trace links become the map which helps engineers navigate through the complexity of product design. The links can be followed, similar to discovering related web pages when browsing. The links can be queried, for example to know how artifacts which are separated by multiple links are connected, similar to asking a navigation system like Google Maps for the route to go from A to B. The links can be commented for improved collaboration, similar to having a discussion thread in social media. The links can support change management by quickly making visible artifacts impacted by a change. The digital thread is meant to support this kind of modern traceability activities. In contrast, current solutions to manage trace links are siloed, and cannot be queried nor analyzed at a global level, in a tool-agnostic way. Current traceability solutions are tied to a specific systems engineering application, PLM solution, or to spreadsheets, and don’t support modern traceability activities.

A tool-agnostic way to achieve the digital thread supporting modern traceability activities is based on standards, especially API standards. With over 500 different applications being used in engineering, more than 500 different application programming interfaces (APIs) exist to access data in these different applications or databases. A higher level of abstraction is required for accessing data independent of its storage solution.

Web APIs have gained a lot of adoption in the last decade. Some say that we live in an API economy. Among Web APIs, REST APIs have become the de facto standard. Even though REST APIs provide a significant level of standardization, many aspects still need to be standardized. REST APIs use different identifiers for data, different schema definitions, different machine-readable descriptions of their web services which is necessary for machine-based discoverability, different ways to describe data versions and updates, and last but not least different ways to describe links between data.

All these aspects can be standardized as shown for example by Open Services for Lifecycle Collaboration (OSLC). Over 50 OSLC APIs have been developed for engineering applications. IBM and Mentor Graphics are large vendors supporting OSLC. Recently, interest in OSLC has grown among smaller vendors who want to take advantage of this new technology. Vendors like

Contact-Software (PLM), Sodius and MID (systems engineering) are creating solutions supporting OSLC. Approaches similar to OSLC exist such as Solid and Hydra. However, these approaches have not yet reached the same level of adoption and maturity as OSLC. It is likely that a mix of these approaches will ultimately gain broad adoption.

Standard APIs are necessary for creating a new kind of application which can work with data from different sources independent of storage location or storage solution. For example, a “Google Search” for data is only possible if data is accessible through a standard API. Current Web search engines can index documents on the Web saved on different servers because they can access documents through a common interface, in this case the HTTP protocol.

A common interface for data decouples application logic from data storage, in the same way that the HTTP protocol decouples Web applications (e.g. search engines, browsers) from document storage. This decoupling allows organizations to mix-n-match applications with datasets as they choose. For example, an organization can easily run the latest AI algorithm on old existing data without having to deal with the traditional difficulties of having to use multiple different APIs to access the data in the first place. Taking into account the high speed of innovation in AI, and the resulting short half-life of AI algorithms, it becomes critical for organizations to quickly benefit from the latest AI algorithms. Standard APIs are disrupting traditional software solutions by breaking apart the traditional proprietary APIs. Standard APIs enable new opportunities for faster and more holistic data analysis. Standard APIs can therefore be considered an important enabler for innovation in organizations undergoing digital transformation efforts.

Following Metcalfe’s Law on the value of networks, the impact of standard APIs is proportional to the square of the number of standard APIs. Trace links as described earlier, which are the foundation for modern traceability activities, can be defined between more data as more standard APIs exist. As more links exist, navigating through links becomes more interesting to engineers.

Engineers may initially define links simply for the purpose of traceability, and over time realize that the links can actually be used for a second purpose, namely as building blocks for the definition of model transformations. In this second use case, the digital thread can help support semantic interoperability. Links can be used to identify semantic correspondences between data elements. As engineers use many different vocabularies related to specific domains, and as these vocabularies continuously evolve, it becomes very hard to keep track of how these different vocabularies relate to each other. Some vocabularies may have semantic overlaps. For example, some vocabularies may share concepts with the exact same meaning but with a different name, in which case a one-to-one mapping would describe the semantic correspondence between these concepts. In some cases, the mapping may be one-to-many. These mappings can be collected in a model transformation in order to automatically translate data conforming to one language into data of another language. Running these data translations from one language to another is useful for the purpose of reuse and synchronization.

It is a challenge to know what the correct mappings between languages are. First, it is challenge of scalability as there are many different vocabularies used in engineering, possibly in the hundreds within large organizations. By trying to find a translation between each possible language, this would require tens of thousands (100x100) of model transformations to be defined. The number of required model transformations can be reduced by using a universal language which is the superset of all possible domain-specific concepts that need to translated. In such an approach, only hundreds of model transformations would need to be defined. If domain-specific languages were static and would not contain additional concepts over time, nor have the meaning of concepts change over time, then the definition of a stable universal language and related model transformations would be a one-time effort. Unfortunately, domain-specific languages continuously evolve requiring continuous updates to the universal language and model transformations.

Another challenge is that different model transformations may exist between different languages. For example, a Modelica model can be mapped into a SysML parametric diagram or a SysML internal block diagram. After 2 years for the Modelica and SysML communities to agree on an official standard for the mapping between Modelica and SysML, many engineers were nevertheless choosing a different mapping from the standard mapping. This means that the definition of model transformations cannot be imposed in a top-down approach, but in practice bottom-up. Based on how engineers agree to use domain-specific languages, and related mappings, model transformations can be defined to support the automatic translation of concepts exactly as intended by engineers. As links can describe a semantic mapping between concepts, or a mapping between instances of concepts, they form the atomic unit of model transformations between languages. Multiple links describing semantic mappings can then be combined into a single model transformation for automatically translating data between different languages.

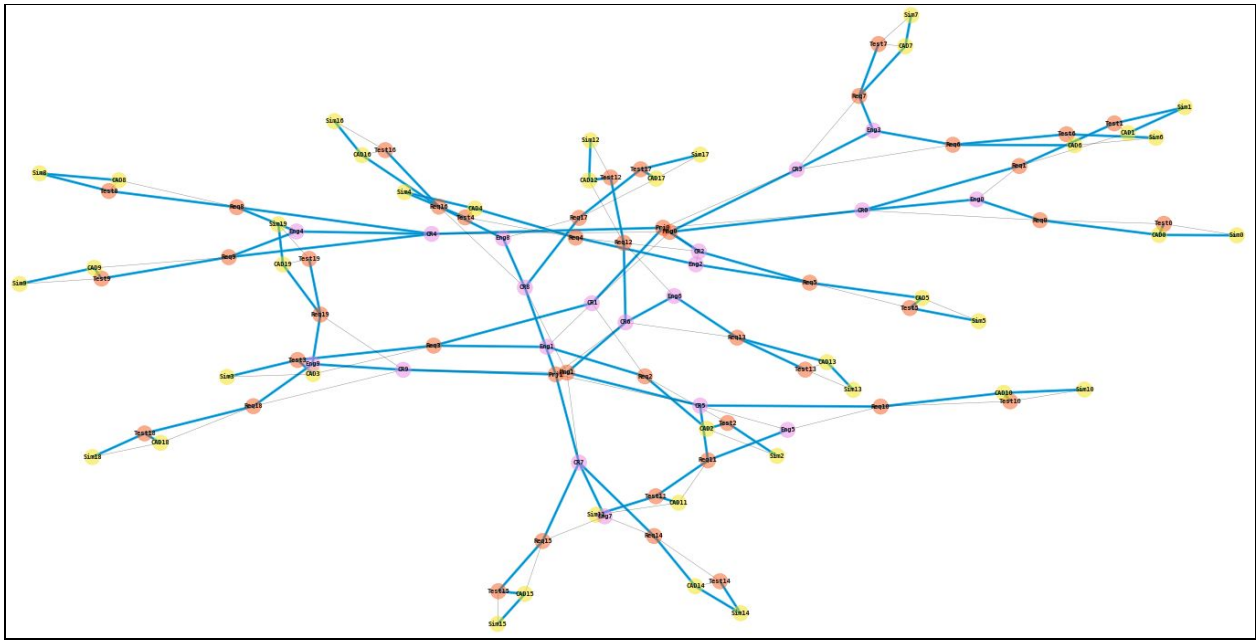
Creating trace links manually can be time-consuming and error-prone. An engineer first needs to find both data elements to be linked. This may require checking the context of data elements to be sure about the right selection of data elements. Even if engineers spend a lot of time creating links, as with any human activity, some of the defined links will be wrong and some will be missing.

Based on patterns in manually defined links, several automatic approaches exist to predict missing links. The investigated approaches were deep learning on graphs, heuristics, and graph mining. Deep learning has gained a lot of popularity in recent years and it is often considered the most advanced machine learning technique. Heuristics based approaches for link prediction are considered old yet reliable. Graph mining is typically not considered for link prediction but it can identify complex patterns, and be used for link prediction even though it requires a lot of computation.

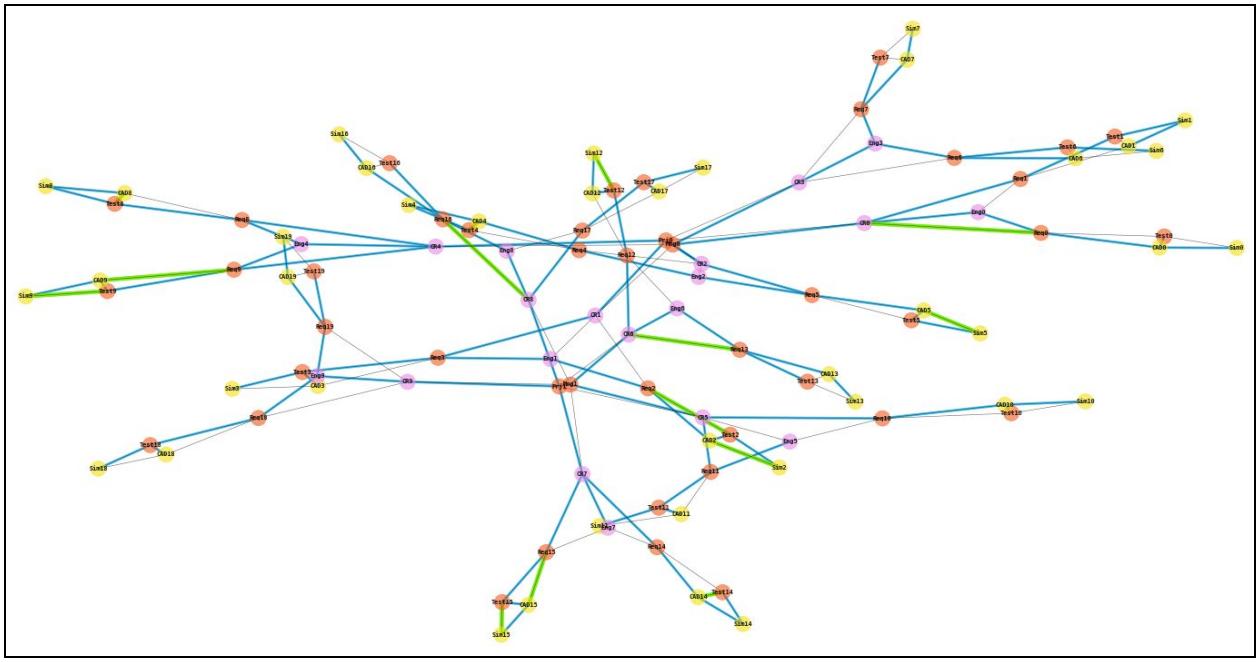
In the academic literature, an increasing amount of papers are published on the application of deep learning for graphs. The papers often cite good prediction results. However, most academic literature inflate the quality of link prediction results by selecting a very specific test dataset for evaluating the link prediction model, namely a dataset which only includes the links to be predicted. In general, a link prediction model receives as input a set of possible links, and for each one, the model will output a boolean true/false value indicating if the link exists or a likelihood between 0 and 1 of the link existence. The choice of dataset to evaluate a link prediction model is critical. It will influence the number of true positive vs false positive predictions. If the model predicts too many false positives compared to true positives, the model becomes useless as the objective of the link prediction model is to help engineers in identifying highly probable links. By recommending false positives, the link prediction model is actually giving more work to engineers and this needs to be avoided. In the academic literature, link prediction models using deep learning are evaluated using a dataset containing only links to be predicted. The link predictions may then be accurate to 80% or even 90%.

However, in reality, a link prediction model needs to be evaluated for a dataset describing all possible links as engineers have no idea which links can be predicted. In that case, the dataset will contain a very high number of link candidates which the model should predict as non-existent. The evaluation results are then very different. The ratio of true positives to false positives predictions is then too low and the model is not considered useful. Another drawback of deep learning on graphs is that it is currently not able to clearly identify complex patterns composed of a chain of multiple nodes in a graph. Most trace links in engineering will follow some complex patterns covering multiple nodes. A second drawback is that trace links used in engineering will form a graph of relatively small size, compared to the size of graphs used to describe social networks. Deep learning only works with a lot of data. The size of engineering graphs, as used in the digital thread, may be too small for deep learning algorithms. It can be concluded that link prediction using deep learning is currently not suitable for graphs as used in the digital thread. However, this assessment may change as a lot of research is currently being performed on the application of deep learning on graphs.

A simpler alternative to deep learning is to use a heuristics-based model for link prediction. Such a model will predict links between nodes, based on nodes having common neighbor nodes. This approach is simple yet it can produce useful results as many patterns in graphs involve nodes having common neighbors. In many graphs, this may be the most common pattern. Heuristics-based link prediction models work on small graphs, and can thus be applied on graphs describing the digital thread.



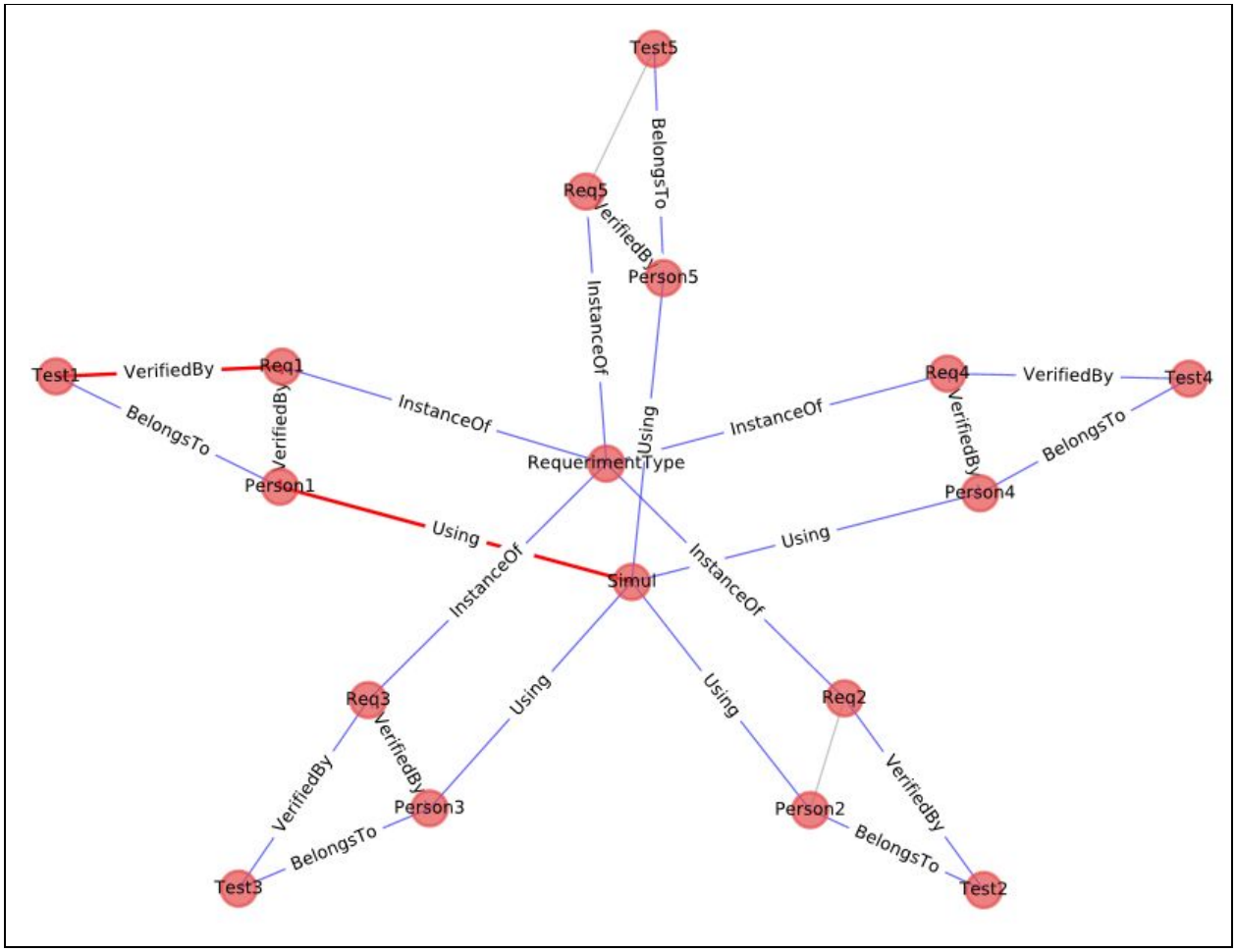
Initial graph for heuristics-based link prediction model. Blue edges are existing known links. Grey edges are the links to be predicted.



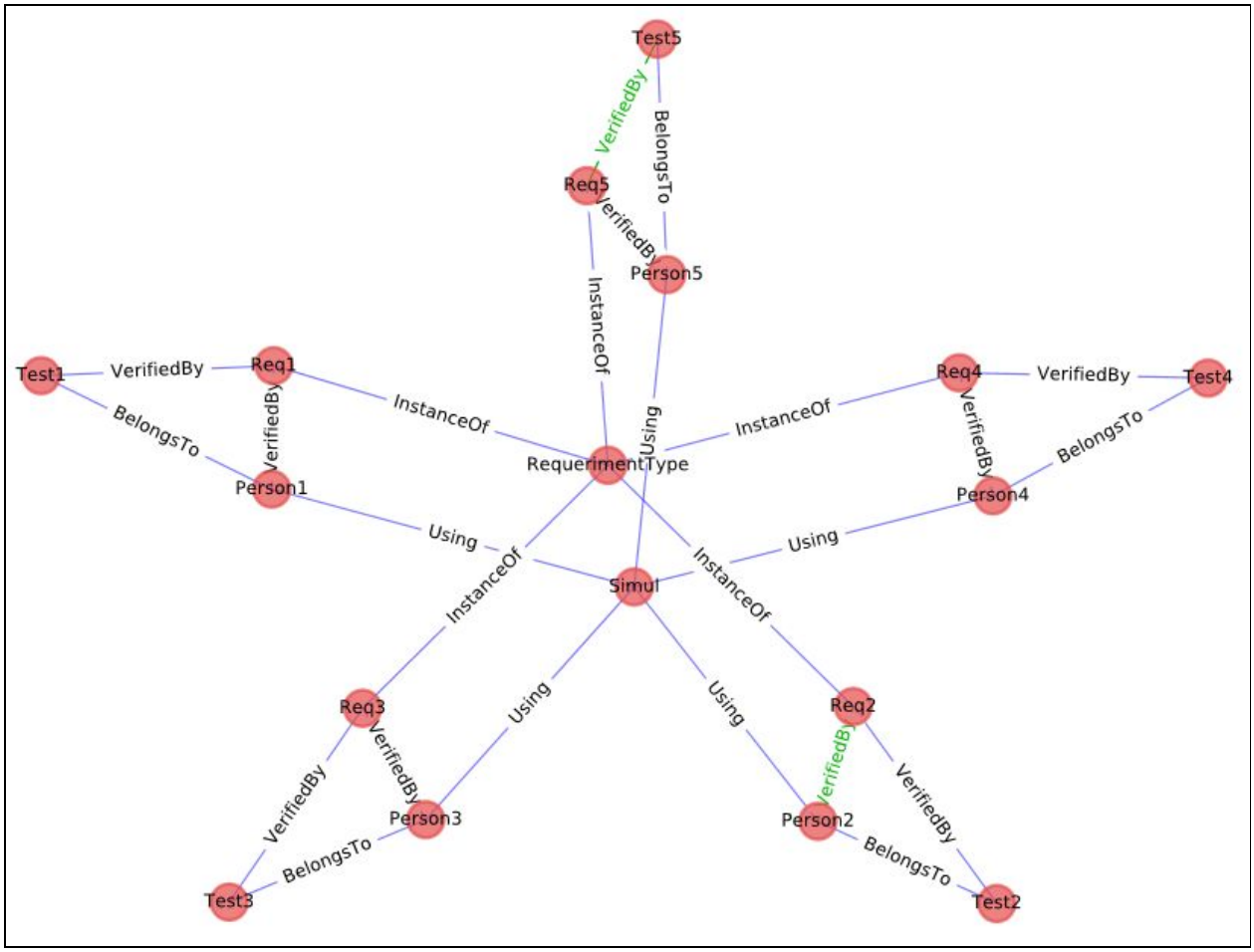
Graph containing links predicted by heuristics-based model. Blue edges are existing known links. Grey edges are the links to be predicted. Green edges are the predicted links

A third alternative is to apply a (semi) brute-force approach to identify patterns in a graph. This is called graph mining. It is computationally-intense but it can identify very complex patterns in a

graph very accurately. After identifying patterns, a link prediction model can then use these patterns to see if additional pattern instances can be found by adding links to the graph. If so, the link prediction model would output these links as probable links. Graph mining can be performed on small graphs. The subsequent activity of trying to find additional pattern instances through additional links is very computationally intense, and ideally requires distributed computing resources, even for small graphs.



Initial graph for link prediction model based on graph mining. The pattern identified through graph mining is shown through red links. It is composed of 4 nodes.



Predicted links are additional links creating additional pattern instances. Predicted links are shown in green.

Conclusion

New ways of connecting data allow engineers to better understand the big picture and to better analyze IoT data. Connecting data requires easy data access, enabled by standardized Web APIs for data sources. Connected data should be viewed as a graph, on which additional analysis can be performed for example for link prediction. Existing approaches for link prediction are not perfectly suitable for graphs used in engineering which are relatively small and have complex patterns. Different approaches for link prediction using deep learning, heuristics and graph mining have been investigated. Better results could be obtained by combining different link prediction approaches, and by taking into account additional information such as string values associated to graph nodes for natural language processing.

References

[DigitalThreadUSAF2013] Why Digital Thread? USAF 2013, https://www.dodmantech.com/ManTechPrograms/Files/AirForce/Cleared_DT_for_Website.pdf

[GE-AI-Future-2016] Four of GE's top engineers talk about business, competition and the future https://www.businessinsider.com/top-ge-engineers-on-business-competition-and-future-2016-10

[Future-of-traceability-2016] Future of traceability, Jama, 2016 https://www.youtube.com/watch?v=2Fp35S2a1gU&list=PLIk9my-nIqejgSWGzm87trLx_3oX4njy6