The Accountability Fabric: A Suite of Semantic Tools For Managing AI System Accountability and Audit*

Milan Markovic[1]✉, Iman Naja[1], Peter Edwards[1], and Wei Pang[2]

[1] Computing Science, University of Aberdeen, Aberdeen AB24 3UE, UK.
{milan.markovic, iman.naja, p.edwards}@abdn.ac.uk
[2] School of Mathematical and Computer Sciences, Heriot-Watt University
Edinburgh, EH14 4AS, UK. w.pang@hw.ac.uk

Abstract. The life cycle of an AI system is a complex multi-stage undertaking that typically involves a range of human stakeholders (e.g., developers, managers, users) who can potentially be held accountable if harm is caused by the system. In this paper, we present the Accountability Fabric, a suite of semantic tools for managing the creation and audit of accountability knowledge graphs.
Demo Link: https://rains-uoa.github.io/ISWC_2021_Demo/

Keywords: AI · Provenance · Accountability.

1 Introduction

The widespread adoption of AI systems and the risks associated with errors, bias and other negative consequences have highlighted the need to regulate such systems, and have led to calls to make them (and their developers) accountable.

To situate our work, we use the definitions presented in our previous work [6], where an AI system comprises 'core AI' components (e.g., a Machine Learning model) plus other supporting ones (e.g., API wrappers); and its life cycle consists of four stages: *Design*, *Implementation*, *Deployment*, and *Operation*. Moreover, an accountable AI system is one which can be inspected, audited, or reviewed with the goals of (*i*) making transparent the processes of each stage of its life cycle; (*ii*) exhibiting compliance with hard laws (i.e. laws and regulations), and soft laws (i.e. standards and guidelines); and (*iii*) facilitating investigations of erroneous decisions or failures and determining the responsible human agents.

In recent years, and in an effort to enhance transparency of ML systems, a number of prominent frameworks have been proposed for recording metadata

* Supported by the award made by the UKRI Digital Economy programme to the RAInS project (ref: EP/R033846/1).

about datasets [2], Machine Learning (ML) model implementations [4], and also more comprehensive system descriptions [1]. However, with the exception of Google's Model Card Toolkit[3], these frameworks only record data in unstructured formats without the ability to manipulate data programmatically, and none capture decisions of human agents that may have influenced the overall AI system (e.g., approval of a design specification by a project manager). Moreover, it is typically not straightforward to collect, record and manage such metadata due to them being generated by various agents (e.g., developers use in-code comments, decision makers use online forms and text reports), across different time frames within the system life cycle, in various siloed locations. We argue that Semantic Web technologies are ideally suited to facilitate both the representation and linking of such information.

In our approach, we utilise PROV [5] to describe information related to the AI system life cycle stages in the form of retrospective linked causal graphs consisting of *activities*, *entities*, and *agents*; we refer to these as *accountability traces*. Furthermore, we argue that to fully realise accountable AI systems, meaningful information is required, the recording of which must be planned. We thus extend EP-Plan [3] to describe *accountability plans* to specify information that should be captured throughout the system's life cycle and to guide the recording of corresponding accountability traces. Accountability plans and traces, together referred to as *accountability information*, form semantic knowledge graphs that can be queried for the information necessary to establish accountability of human agents.

2 System Overview

The *Accountability Fabric* is a suite of tools that utilises a provenance-based approach for recording and querying accountability information using semantic graphs. The tools are supported by a Spring Boot[4] back end server application utilising the RDF4J library[5] to communicate with the GraphDB[6] repository which stores accountability information. User interfaces are built using HTML, CSS and Javascript. The fabric is supported by four main ontologies: PROV-O[7], EP-Plan[8], SAO[9], and RAInS[10]. SAO defines the mechanisms for representing accountability plans and their corresponding accountability traces, and RAInS extends SAO to describe the life cycle of AI systems with Machine Learning components. The reader is referred to [6] for more details on how these ontologies are integrated.

[3] `https://github.com/tensorflow/model-card-toolkit`
[4] `https://spring.io/projects/spring-boot`
[5] `https://rdf4j.org/`
[6] `https://www.ontotext.com/products/graphdb/`
[7] `https://www.w3.org/TR/prov-o/`
[8] `https://w3id.org/ep-plan`
[9] `https://w3id.org/sao`
[10] `https://w3id.org/rains`

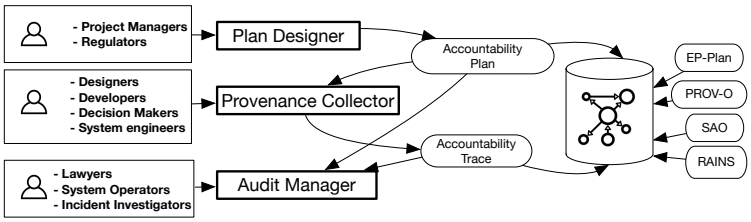

Fig. 1. Accountability Fabric Overview.

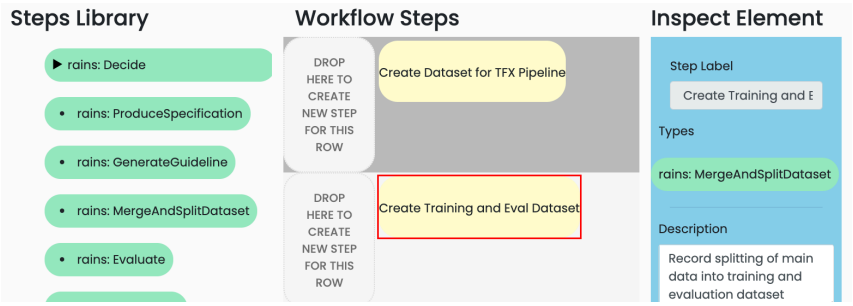

Fig. 2. Partial Screenshot of *Plan Designer* UI.

Figure 1 illustrates three main components: *Plan Designer*, *Provenance Collector*, and *Audit Manager*, along with examples of potential users. The *Plan Designer* supports creation of accountability plans for each life cycle stage to define what accountability information should be recorded (Figure 2). Plans are described as workflows that consist of steps representing planned activities (e.g., implementation of an ML model). Such steps also include inputs and outputs which represent high level references to the type of information that should be collected (e.g., information about the dataset or ML model, or previously made decisions). Steps may include additional metadata such as constraints, which are evaluated against the accountability trace (e.g., the model description must include a list of its limitations).

The *Provenance Collector* supports both manual and semi-automatic creation of accountability traces. For manual input, a user would use an interactive Web form (Figure 3A), the automatic generation of which is guided by the relevant accountability plan. The fabric also provides a (Python) script for integrating data recorded by the Model Card Toolkit and executed in a Colab/Jupyter Notebook environment[11] (Figure 3B). This converter also has the ability to visualise any violations of plan constraints, defined using SHACL[12]. These can be used, for example, to indicate to the developer that certain information should have been provided. The *Audit Manager* (Figure 4) provides an interactive Web

[11] `https://colab.research.google.com`
[12] `https://www.w3.org/TR/shacl/`

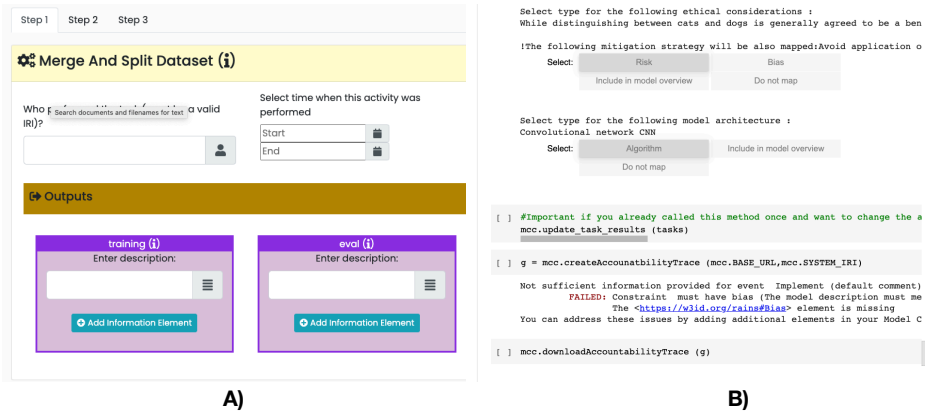

Fig. 3. A) Partial screenshot of manual interface for creating *accountability traces*. B) Partial screenshot of Colab notebook running the script for converting data from the Model Card.

interface for exploring accountability knowledge graphs. Several visualisation options are provided including the report-like visualisations inspired by Model Cards [4].

The RAInS ontology has recently been extended to cover the implementation stage in addition to the design stage described in [6]. The ontology has also been extended with OWL constraints, which are used by the *Plan Designer* to guide the user during the creation of *accountability plans*.

3 Demonstration

All main *Accountability Fabric* components are demonstrated, namely the *Plan Designer* (Figure 2), *Provenance Collector* comprising the manual accountability trace input UI and the mapping tool for integrating data captured using the Model Card toolkit, and the *Audit Manager* for exploring accountability graphs (Figure 4). The demonstration will take the participants through the process of planning and recording accountability information for an example AI system. The aim is to demonstrate three key capabilities of the *Accountability Fabric*: planning, recording, and auditing.

4 Conclusions

The *Accountability Fabric* showcases how semantic technologies can be used to support the accountability of AI systems by collecting, integrating, and visualising information describing the design and implementation stages of an AI system life cycle; with other life cycle stages due to be incorporated in the near future. Our future work will focus on evaluating the individual components of

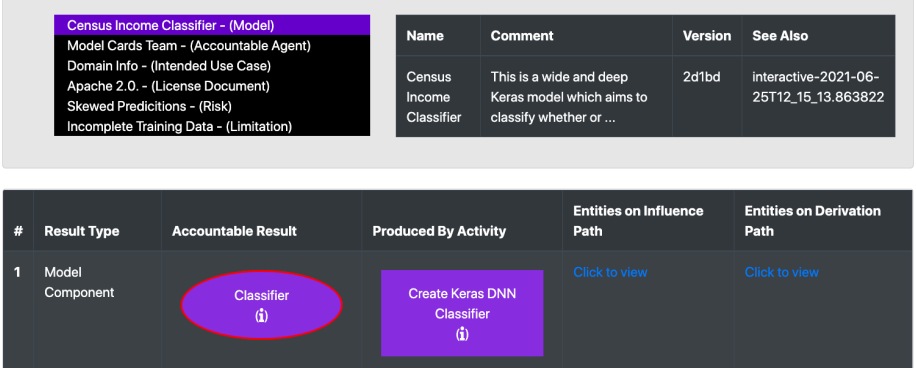

Fig. 4. Partial Screenshot of *Audit Manager* UI.

the *Accountability Fabric* (see Figure 1) with users such as AI system developers, project managers, and lawyers; we will also compare the suite's capabilities against existing text-based approaches such as Data Sheets [2], Fact Sheets [1], and Model Cards [4].

References

1. Arnold, M., Bellamy, R.K., Hind, M., Houde, S., Mehta, S., Mojsilović, A., Nair, R., Ramamurthy, K.N., Olteanu, A., Piorkowski, D., et al.: Factsheets: Increasing trust in ai services through supplier's declarations of conformity. IBM Journal of Research and Development **63**(4/5), 6–1 (2019)
2. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J.W., Wallach, H., Daumé III, H., Crawford, K.: Datasheets for datasets. arXiv preprint arXiv:1803.09010 (2018)
3. Markovic, M., Garijo, D., Edwards, P., Vasconcelos, W.: Semantic modelling of plans and execution traces for enhancing transparency of iot systems. In: 2019 Sixth Int. Conf. on Internet of Things: Systems, Management and Security (IOTSMS). pp. 110–115. IEEE (2019)
4. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I.D., Gebru, T.: Model cards for model reporting. In: Proceedings of the conference on fairness, accountability, and transparency. pp. 220–229 (2019)
5. Moreau, L., Groth, P., Cheney, J., Lebo, T., Miles, S.: The rationale of PROV. Web Semantics: Science, Services and Agents on the World Wide Web **35**, 235–257 (2015)
6. Naja, I., Markovic, M., Edwards, P., Cottrill, C.: A Semantic Framework to Support AI System Accountability and Audit. In: The Semantic Web. ESWC 2021. pp. 160–176. Springer, Cham (2021)