Towards automated provenance collection for experimental runs of agent-based models[*]

Doug Salt[1][0000−1111−2222−3333] and Gary Polhill[2,3][1111−2222−3333−4444]

The James Hutton Institute, Craigiebuckler Aberdeen AB15 8QH Scotland
doug.salt@hutton.ac.uk
http://hutton.ac.uk

Abstract. The abstract should briefly summarize the contents of the paper in 15–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

Replication of social simulation results has been highlighted as a significant issue for the ABM for a number of years (e.g. [2]), and in particular the paper that forms the basis of this work [3]. The focus of replication work has previous just been on the model itself, but as was shown in ibid, the analysis of the outputs of the model can potentially be just as complex, and no less difficult to replicate unless adequate records are kept. Indeed there as an additional attempt to reproduce the results and even with the lessons learnt from ibid, it still took months to recreate the final diagrams that were submitted to paper. The TRACE protocol [4, 1] provides some guidance highlighting the need to keep a notebook of the analysis done and a standardised approach to making that notebook; however, the lessons learned from the replication exercise in [3] show that more detailed guidance on the information that should be recorded. and on tools that could be used to support the process. Ideally the aim should be to completely automate the process, record accurately sources of data used, version check applications used essentially providing a complete graph from data to result.

The output analysis replication in this paper concerns earlier work with FEARLUS-SPOMM, which is a coupled agent-based model of agricultural decision-making and species stochastic patch occupancy metacommunity model that has been used to explore incentivisation strategies to improve biodiversity (Polhill et al. 2013; Gimona and Polhill 2011). Belonging to the 'typification' class of social simulations adfa, p. 1, 2011. © Springer-Verlag Berlin Heidelberg 2011 ("theoretical constructs intended to investigate some properties that apply to a wide range of empirical phenomena that share some common features" – Boero and Squaz- zoni, 2005). This work involved the analysis of the outputs from around

[*] Supported by The Scottish Government

20,000 runs of the model using a number of techniques aimed at demonstrating nonlinearities in the relationship between incentivisation and biodiversity outcome. Recording workflow data on the process used to create analysis can be challenging, and currently there are no codified standards as to how this should be done for ABMs. For FEARLUS-SPOMM, the methods used drew heavily on statistical techniques available as R packages that are as part of core R functionality. Although R allows transcripts of interactive terminal sessions to be saved, the work involved great deal of exploration of different ways of attempting to visualise and analyse the nonlineari- ties in the model results, not all of which were likely to be reported in the paper. Such logs are therefore not the best way to record the means by which the outputs were analysed, and hence the strategy used was to save each analysis or visualisation in a(n R) script. Since the output from the (Swarm) software that generated the output data being analysed used a mixture of text formats, some Perl scripts were also written to process that output into a CSV file for easy import into R. When the MIRACLE pro- ject (Parker et al. 2015) provided a context in which the replication of that analysis was necessary, an opportunity was created to test the viability of the above strategy.

In the rest of this paper, we describe a tool for automatically recording metadata, which can be incorporated into the analysis replication process, and how this was used to regenerate some of the figures in Polhill et al. (2013) and we test a tool to support keeping the records needed to per- form replication more easily, before making some recommendations in concluding remarks.

2 Method

One of the artefacts of the MIRACLE project [?] was the Social Simulation REplication Interface or SSREPI. This is the schema shown in 1. This schema has evolved since it original conception. The newest version of this document may be found here [?]. This is a database schema derived from the Dublin Core [?], the standard XML datatypes [?] and the PROV-O ontologies [?]. The schema is designed with agnosticisim towards the underlying database technology and as such has been implmented both in PostgresSQL [?] and Sqlite3 [?].

There are two important dimensions of distinction to this schema. First is fine- versus coarse- grained metadata. Second is provenance versus workflow. Coarse-grained metadata describes how particular files come (or came) into being, or were (or could be) used to bring other files into being. Fine-grained metadata describes specific values recorded in social simulation outputs. To make the distinction concrete, suppose a simulation produces a CSV file. The data within the CSV file are covered by fine-grained metadata, whilst the fact that the simulation produces the CSV file is coarse-grained. Turning to the other dimension, provenance metadata describes what actually happens (run W of simulation X produced output file Y), whilst workflow metadata describes what could happen (simulation X produces an output file of type Z). The distinctions are summarised in 2 .

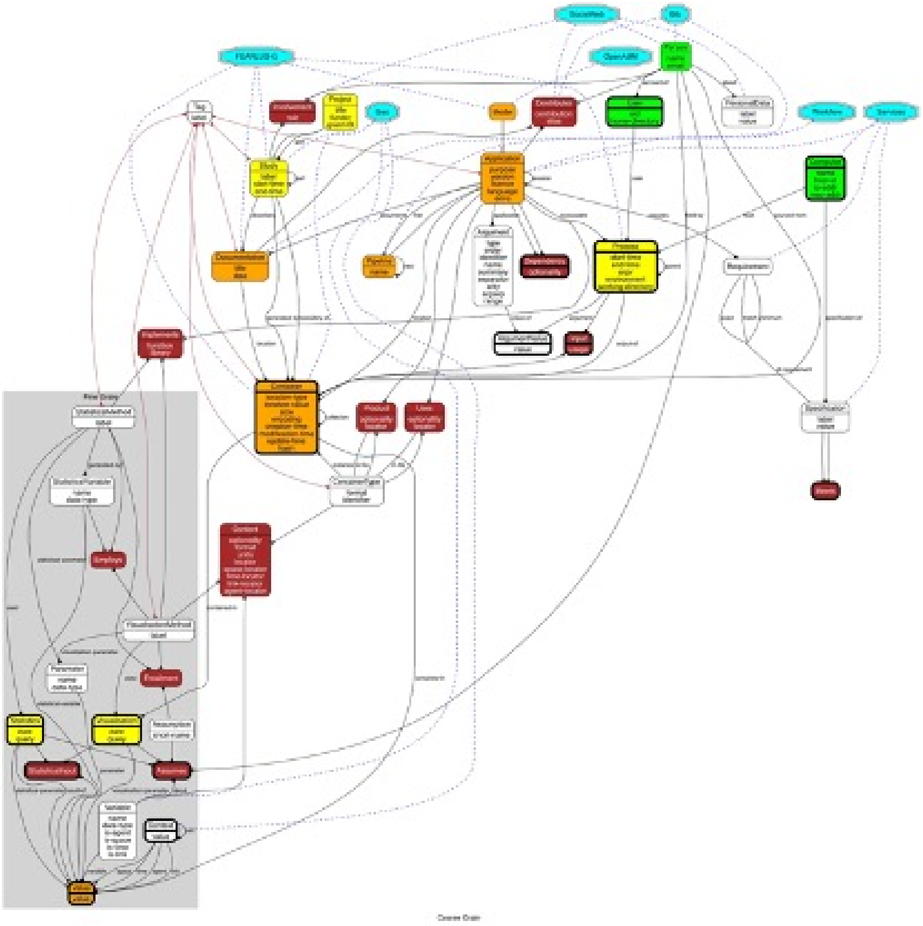

Fig. 1. SSREPI Schema

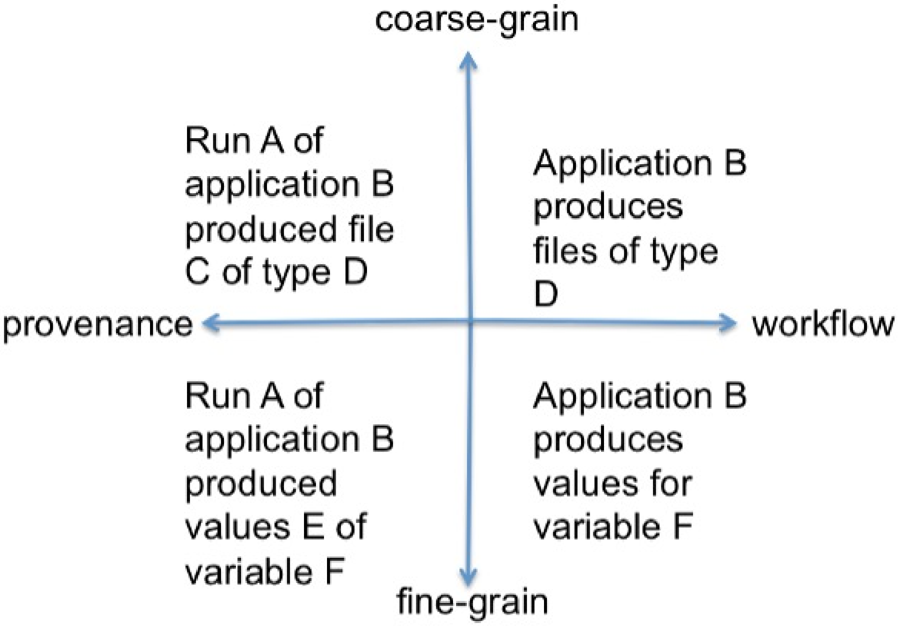
coarse-grain
Run A of application B produced file C of type D
Application B produces files of type D
provenance
workflow
Run A of application B produced values E of variable F
Application B produces values for variable F
fine-grain

Fig. 2. Fine grain vs coarse grain provenance

For the purposes of this short paper we are concentrating on a demonstration of recording provenance. Indeed we use the example, mentioned in the introduction of [?], and modified the original Bash [?] scripts to include the what we denote as *primitives* . So an example of the code

In order to record provenance we We coded the following primitives from the SSREPI interface definition [?]. These are implemented in a mixture of Python [?] and Bash [?]

Broadly speaking SSREPI_application, SSREPI_run, SSREPI_batch, SSREPI_input, SSREPI_output and SSREPI_argument are responsible for recording coarse grain provenance. SSREPI_value, SSREPI_visualisation_variable_value, SSREPI_statistical_variable_value, SSREPI_run and SSREPI_batch record fine-grain provenance. The remaining primitives are largely about recording metadata.

Elaboration of all this may be found in the man pages and design documents in the public repository at git@github.com:DougSalt/ABM-metadata.git

3 Results

Obviously with complicated diagrams result from the 20,000 runs. The challenge becomes how to visualise these. There are several graphs produced so far, these being

Table 1. Table captions should be placed above the tables.

Primitive	Type	Purpose
SSREPI_require_minimum	Metadata	Lower bound on software hardware required
SSREPI_require_exact	Metadata	Exact bound on software hardware required
SSREPI_application	Provenance & Metadata	specifies some executable
SSREPI_me	Provenance & Metadata	Determines executable being run or returns a proper reference to the executable being run.
SSREPI_run	Provenance & metadata	Blocking invocation of an executable which will allow the specification of all inputs, outputs and arguments. Creates run-time provenance information as well
SSREPI_batch	Provenance & metadata	Non blocking invocation of an executable which will allow the specification of all inputs, outputs and arguments. Creates run-time provenance information as well
SSREPI_argument	Provenance	An argument type to an exectuable
SSREPI_output	Provenance	An output type from an executable
SSREPI_input	Provenance	An input type for an exectuable
SSREPI_hutton_person	Metadata	Uses our institutions databases to populate metadata for a given individual
SSREPI_person	Metdata	Provide metadata for a particular actor within this system
SSREPI_project	Metadata	Specifies a project which contains all studies
SSREPI_study	Metadata	A set of experiments makes up a single study
SSREPI_set	Metdata	Sets the default licence and other metadata
SSREPI_involvement	Metadata	Links personnel to a study
SSREPI_paper	Metadata	A paper associated with this study
SSREPI_make_tag	Metadata	Used for building a folksonomy
SSREPI_tag	Metadata	Used to tag any entity with a folksonomy tag
SSREPI_contributor	Metadata	A person with some kind of relation to an executable or script.
SSREPI_statistical_method	Metadata	A statistical method is an approach to computing some statistics. It may be implemented in or as part of an application. A statistical method generates one or more statistical variables as its results, and may use the results of another statistical method in its computation. For example, computing the standard deviation of some data uses the mean of those data.
SSREPI_visualisation	Metadata	A visualisation is the process of creating an image to depict one or more (typically more than one) values. The results of a visualisation appear in a container.
SSREPI_statistics	Metadata	Statistics are activities that compute and populate the values of statistical rvariables. They operate on raw data that are retrieved from the values using a query. To replicate a set of statistics, the query can be rerun, selecting values that are pointed to by containers entries
SSREPI_visualisation_method	Metadata	This describes methods for generating visualisations, which then may appear in the content of a container produced by a pro-

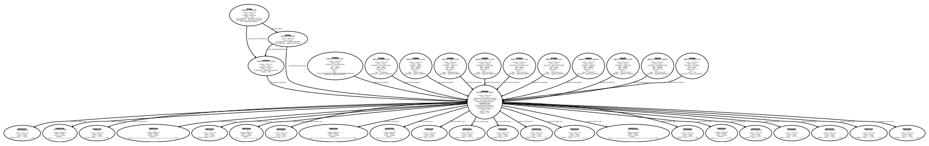

Fig. 3. The workflow sub-graph

We show the workflow graph in fig. **??** and a section of the proveance graph in fig. 4

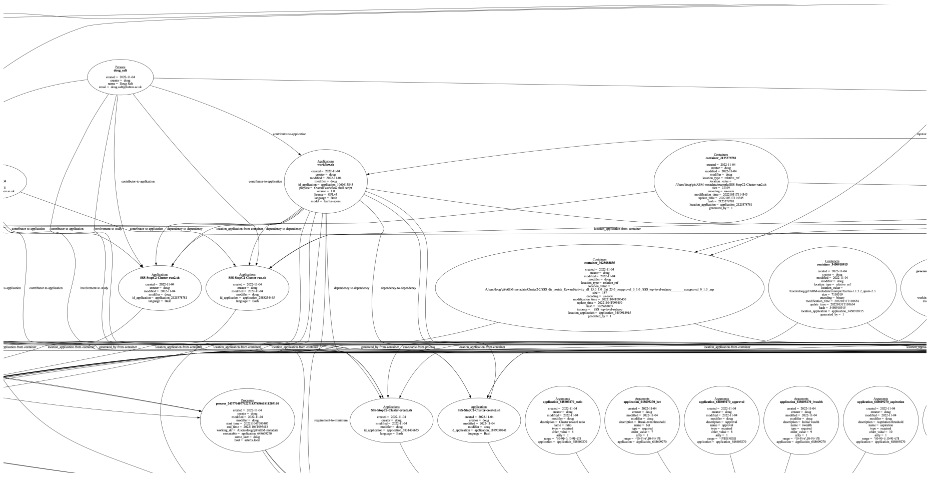

Fig. 4. A small sub-section of the proevenance graph

4 Discussion

What we have learnt from this This is a lot of pfaff. and is still not oven ready

So what use is this provenence meta data. Tracing bad data.

We demonstrate this by tracing a series of bad data across the diagrams.

Eventually we should like to take several databases run some machine learning across them to see if there are any commonalities in experiment set up and post processing.

Additionally we want to produce modules in other languages as well, such as R, python and Julia. Especially the latter.

References

1. Ayllón, D., Railsback, S.F., Gallagher, C., Augusiak, J., Baveco, H., Berger, U., Charles, S., Martin, R., Focks, A., Galic, N., et al.: Keeping modelling notebooks

with trace: Good for you and good for environmental research and management support. Environmental Modelling & Software **136**, 104932 (2021)

2. Edmonds, B., Hales, D.: Replication, replication and replication: Some hard lessons from model alignment. Journal of Artificial Societies and Social Simulation **6**(4) (2003)
3. Polhill, G., Milazzo, L., Dawson, T., Gimona, A., Parker, D.: Lessons learned replicating the analysis of outputs from a social simulation of biodiversity incentivisation. In: Advances in Social Simulation 2015, pp. 355–365. Springer (2017)
4. Schmolke, A., Thorbek, P., DeAngelis, D.L., Grimm, V.: Ecological models supporting environmental decision making: a strategy for the future. Trends in ecology & evolution **25**(8), 479–486 (2010)

keyword=github.com/DougSalt/MABS2023]global.bib