Barriers and Model Curation Issues Associated with Rapid Adaptation of Empirical Legacy ABM in Response to Urgent Policy Maker Queries.

Matt P. Hare [0000-0002-8508-0884], Nick Roxburgh [0000-0002-7821-1831], Doug Salt [0000-0001-5186-9388], and Gary Polhill [0000-0002-8596-0590]

The James Hutton Institute, Craigiebuckler, Aberdeen, AB14 8QH, Scotland
matt.hare@hutton.ac.uk

Abstract. In this paper, we present the methodology and results of the first test of the front-end of a larger protocol guiding a participatory modelling process to rapidly adapt empirical legacy ABMs to effectively respond to policy queries. The ultimate goal of the research is to create a process that could be completed in timespans of weeks rather than several months. The five stages of the front-end protocol tested here cover the formal elicitation of the policy query, the co-construction with the policy maker of requirements for an adapted ABM to answer it, and the process leading to the re-coding of that model. The protocol is designed for the involvement of four actors: a policy maker; a knowledge engineer; a modeller; and a programmer.

The paper concludes with a description of the key barriers identified by the test. These barriers include problems of re-running the legacy ABM from existing code due to changes in software versions since its original development; differences in coding styles between the original developers and the modelling team charged with its adaptation; limits to the descriptive power of ODD; dataset confidentiality and differences in data processing methods; time pressures resulting in design/coding decisions that might lead to errors or model artefacts, etc. We suggest investment in professional model curation as a way to overcome these barriers. The goal of which would be the identification, preparation, and maintenance of legacy ABMs in readiness for their rapid adaptation by modelling teams to meet future policy maker requests.

Keywords: empirical ABM; policy support; participatory modelling; rapid model adaptation; model curation.

1 Introduction

This paper reports on the early stages of a work package, from a large-scale modelling research project funded the Scottish Government, which seeks to answer the research query:

- How quickly can an empirical legacy ABM be adapted to produce a valid model to answer policy makers' urgent policy queries?

The frequent mismatch between research models and what policy makers need for evidence-based decision making is already well documented and stretches backs many years (see for example [1]). The reuse of legacy models is especially unlikely outside their original purpose [2]. Whilst developing trust in the use of ABM is a work in progress [3], participatory modelling approaches have not yet provided the complete solution to this mismatch as revealed by the lack of demonstrable impact on decision making by 60 published participatory modelling exercises between 2003 and 2017 [4]. When it comes to rapid response modelling, the problems are somewhat greater since there is also an obvious tension between the speed at which models can be developed to a suitable level of validity and the speed at which urgent policy answers are needed to emerging opportunities and threats demand, c.f. Covid-19 [5].

Our research seeks to develop a protocol for a participatory modelling process (supporting active participation in multiple phases from query specification and co-design to model use [anonymised reference]) that can be used to rapidly adapt legacy ABMs to act as "good policy advice models" [5] with respect to answering policy makers' urgent policy queries. By "legacy" we mean an ABM that has been built in the past by modellers who might or might not be part of the organization that is seeking to adapt it. In this paper, we describe the methodology and testing of the front-end part of this protocol, covering the elicitation of the policy query, the co-construction with the policy maker of requirements for an adapted ABM to answer it, and the stages up to and including the initial re-coding of that model. The approach we present contrasts to other approaches which attempt to accelerate policy model development. Such alternative approaches may focus on facilitating rapid "informal" coupling of component sub-models via stakeholder participation [6]; improving the speed of access to empirical data by reusing open data, for example at global scale levels, to replace missing local scale data needed for humanitarian disaster responses [7]; or on developing modelling shells or toolkits containing pre-existing model components or agents that can be easily recombined for bespoke modelling activities [8, 9].

2 The Front-End Protocol

Our front-end protocol as it currently stands includes seven process stages and is designed for the involvement of four actors: the policy maker; the knowledge engineer; modeler; and programmer (c.f. Section 2.12, Fig.2 in [10]). Inspired by Castro et al.'s six-step[1] mapping-mediated, knowledge engineering process for ontology development [11], the seven stages (Fig. 1) are:

- Stage 1 – the modelling team receives from the policy maker makes an urgent request for support in answering a new, emerging policy query;
- Stage 2 – the modelling team identify an appropriate legacy ABM that could be redesigned and repurposed to respond to the query;

[1] Castro's six steps were: purpose identification; reusable ontology identification; domain analysis and knowledge acquisition; iterative informal ontology development; formalization of the ontology; and evaluation. Our protocol does not include "evaluation" yet.

- Stage 3 – the knowledge engineer uses the ODD (Overview, Design concepts and Details) [12] documentation of the legacy ABM to develop a core causal loop model [13] of it;
- Stage 4 – the knowledge engineer interviews the policy maker and uses the causal loop model in two ways, i.e., as: i) a medium of model communication - to present an overview of the ABM which includes an explanation of its key ontology and dynamics; and ii) a contrived knowledge elicitation method [14] for eliciting both explicit and tacit knowledge from the policy maker - to facilitate the policy maker to expand the core causal loop model in order to identify and agree upon the new ontological elements and causal relationships required for the model to answer their policy query.
- Stage 5 – the knowledge engineer converts these requirements into a formal requirements specification document for the target ABM;
- Stage 6 – the modeller converts the requirements specification document into a design specification explaining how the legacy ABM needs to be adapted, and identifying any new data sets that should be used;
- Stage 7 – the programmer uses the design document to then recode and adapt the target ABM.

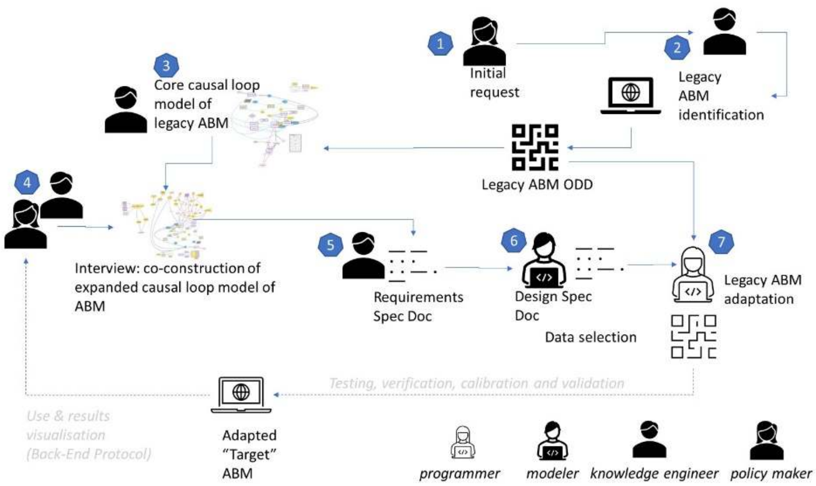


Fig. 1. The flow of 7 stages in the current front-end protocol within a larger participatory modelling process. Light grey lines indicate parts of the protocol still to be developed, including code testing, verification, validation and results visualisation.

3 A First Test of the Protocol

The first test of the front-end protocol involved adapting a legacy ABM developed at the Authors' organisation but whose lead developer now works elsewhere: RISC [15]. This empirical ABM models the impact of different Brexit scenarios on the future number of small, medium and large cattle farms in Scotland. Very broadly, decisions to expand, contract or maintain herd sizes by farms are made according to different decision models depending on the type of farm. Most relevant to the examples in this article is that some farms' decisions are based on their own or other farms' calculations of current profit (for a description of the other decision models, see Section 3 in [15]).

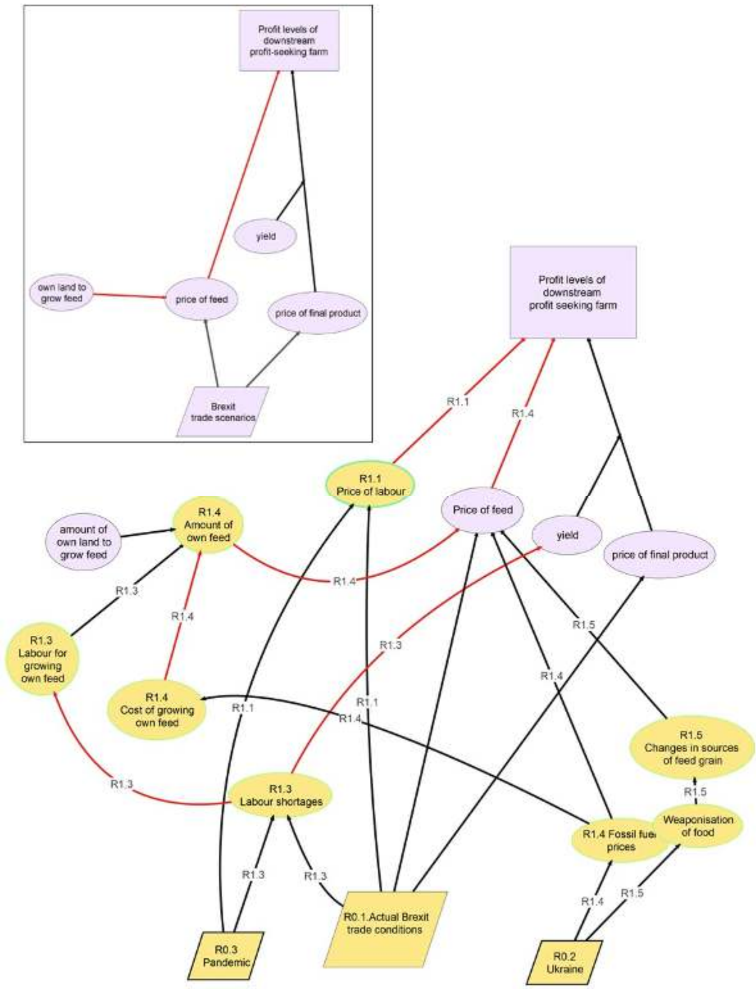


Fig. 2. Top inset: A layer of the core causal loop model, based on the ABM's ODD, related to farm profit calculations as presented and explained to the "policy maker". **Bottom**: Core model extensions (enumerated yellow nodes and arrows) co-constructed with the "policy maker" during the interview. Black arrows signify directed, directly proportional relationships between two nodes; red ones, inversely proportional relationships. Causal loop models visualised in VUE [16].

A mock participatory modelling process was set up whereby the role of knowledge engineer, modeller and programmer were played by staff from the organisation who had had no previous involvement in the model (authors 1, 2 and 3, respectively) but were skilled in their respective roles. Author 4, due to their working relationship with the legacy ABM and consequent knowledge of both the domain and model (although not the main developer), played the role of "policy maker" (for a fictitious "Cow Policy Unit") but was a scientist, unskilled in that role. The first three authors had no prior warning as to what the fourth author would say in their interview, other than the fact that the latter would want adaptations to the ABM. It is for this reason that our mock process began at Stage 3, continued to Stage 7 inclusive. In calendar days, this first test took two months to complete representing approximately 15 person days of work.

3.1 Stage 3 – Development of the Causal Loop Model.

The ABM's ODD was used by the knowledge engineer to develop a systems representation of the ABM in the form of a core causal loop model. This model was developed in several *layers* representing different parts of the model. The layer of this core model that represents the calculation of farm profit levels is illustrated in Fig. 2 (Top inset).

3.2 Stage 4 – Interview with "Policy Maker" and Co-Construction of Extensions to the Causal Loop Model

The core model was presented and explained, layer by layer, at the start of the meeting with the "policy maker". The process of eliciting the policy query and model extensions required to answer it was carried out via co-construction of an extended version of that causal loop model with the "policy maker" (see Fig 2, Bottom, for the elicited extended causal loop model). The method of co-construction was driven using a semi-structured interview script that was adapted in real time during the interview to link to the policy query and the ongoing adaptation of the causal loop model.

The interview questions began with asking the "policy maker" an initial question about the key issues currently impacting the cattle sector that were of major concern. Their responses *actual Brexit trade conditions*, *the war in Ukraine* and *the pandemic* were added to the bottom of the causal loop model, represented in Fig. 2 (Bottom) by the yellow boxes R0.1, R0.2, and R0.3 respectively. Next, new ontological components of the model were identified by asking a follow up question: what they thought were the most important impacts of each of the key issues on the sector. Whilst the "policy maker" answered, the causal loop model was adapted by the knowledge engineer in real time, whilst seeking the former's approval for those changes. An example of such incremental co-construction of the model is the case of R0.2 *war in Ukraine* (see Fig. 3). When the "policy maker" answered the follow up question, above, by responding "the war in the Ukraine raises the price of fossil fuels", the resulting contemporaneous change in the model was the addition of nodes R0.2 "Ukraine" and R1.4 "fossil fuel prices" and their connection by a directed black arrow, R1.4, signifying that the former causes an increase in the latter (Step 1, Fig. 3).

Development of the model continued through a forward-chaining query selection strategy analogous to the forward-chaining deductive processes employed in expert system inference engines. That is, follow up questions were posed to the "policy maker" which were equivalent to "if X is known, then what happens as a result?". In this case, once the concept of "fossil fuel price rises" appeared in the model, the following question was then asked by the knowledge engineer: "what is the impact of fossil fuel price rises?". The answer that "it increases the price of feed" produced a new R1.4 arrow linking fossil fuel prices to the existing model node "price of feed" (Step 2, Fig. 3), already linked to the farm profit levels calculation.

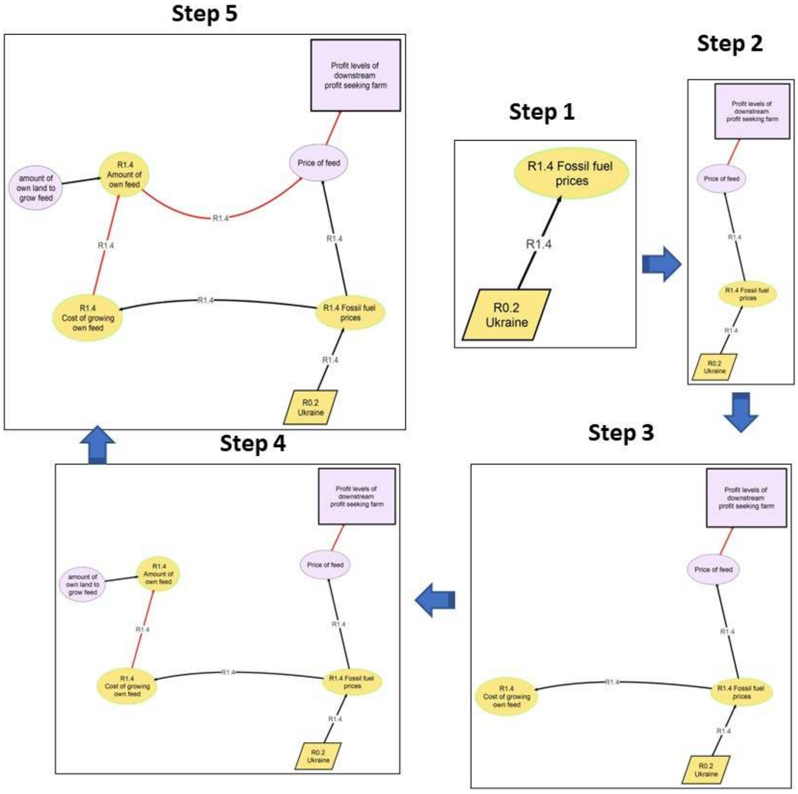


Fig. 3. Step-by-step incremental adaptations to the core causal loop model (all enumerated yellow nodes and arrows) with respect to the calculation of farm profits resulting from the interview (purple nodes and other arrows are from the core model). Alphanumeric Rx.y indicates the related specific requirement in the Requirements Specification documentation (see Stage 5 below).

When the "policy maker" then subsequently added that fossil fuel price rises also made the cost of growing their own feed increase, a new connection leftwards was made from the fossil fuel price node to a new R1.4 node "cost of growing their own feed" (Step 3, Fig. 3). Further forward chaining queries from this new node resulted in the creation of a R1.4 node "amount of own feed" influenced by both an original node in the model "amount of own land to grow feed" and the new node "cost of growing their own feed" (Step 4, Fig. 3). Finally, the link to profit calculations via the cost of feed was made when the "policy maker" identified that when the amount of own feed is high, the cost of [bought-in] feed decreases (Step 5, Fig. 3). A similar second and third tranche of forward-chaining queries were then asked in order to adapt the causal loop model to include both new performance indicators and policy instruments they wanted to test. By the end of the interview (duration 1 hour), the user story, main policy query and sub-queries had been identified based on the extensions to the causal loop model (see Fig. 4, Top left).

3.3 Stage 5 – Requirements Specification

The adapted causal loop model, supported by interview notes, was used to write a requirements specification document (Fig. 4, Top right and Top left) based on an agile project management [17] template[2]. Five requirements for modifications to existing model calculations, three new performance indicators and three new policy instruments were documented.

3.4 Stage 6 – Design Specification

The modeller used the requirements specification document (including the core and extended causal loop models), the ABM's ODD and the original code to specify a design for the programmer to follow to recode the legacy ABM. Fig. 4, Bottom, shows the level of description provided by the modeler for requirement R1.4.

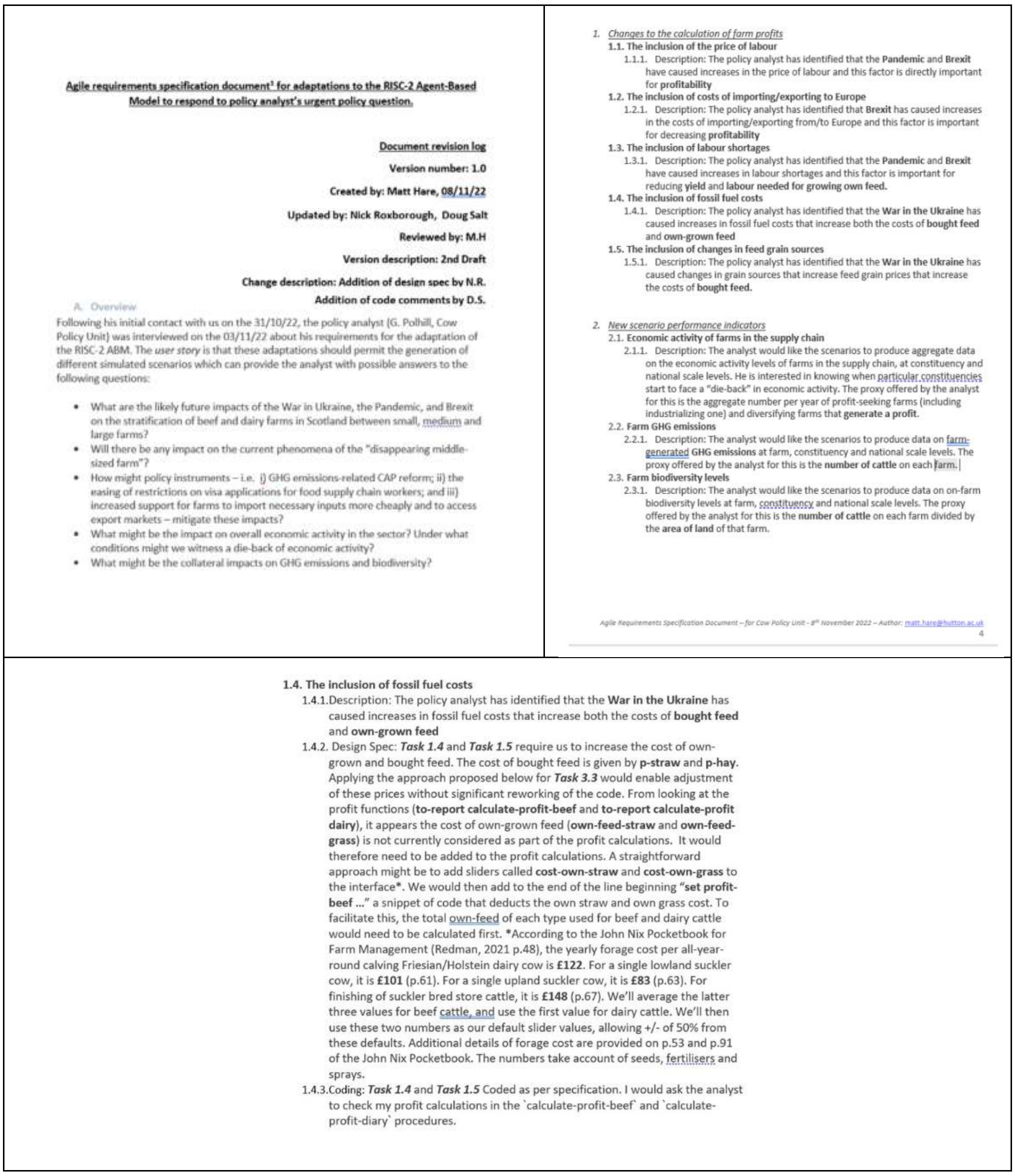
3.5 Stage 7 - Coding

The programmer used the design spec and original documented code to adapt the legacy ABM code (Fig. 5). On completion, they added their comments in Section 1.4.3 of the developing requirements and design document (Fig. 4, Bottom).

4 Conclusions

In this paper, we have presented the methodology and outputs of the first test of five of seven stages of a larger protocol guiding a participatory modelling process to rapidly adapt empirical legacy ABM to effectively respond to policy queries. The goal is to find out what might be needed to be done to complete such a process in timespans of weeks rather than several months. The testing of this part of the front-end protocol and the feedback discussions between the four "actors" involved in it, revealed the following issues affecting the possibility of rapid adaptation of legacy ABM *even before the critical processes of model verification, validation and participatory use are considered*:

[2] See template provided by https://reqtest.com



Agile requirements specification document[1] for adaptations to the RISC-2 Agent-Based Model to respond to policy analyst's urgent policy question.

Document revision log

Version number: 1.0

Created by: Matt Hare, 08/11/22

Updated by: Nick Roxborough, Doug Salt

Reviewed by: M.H

Version description: 2nd Draft

Change description: Addition of design spec by N.R.
Addition of code comments by D.S.

A. Overview

Following his initial contact with us on the 31/10/22, the policy analyst (G. Polhill, Cow Policy Unit) was interviewed on the 03/11/22 about his requirements for the adaptation of the RISC-2 ABM. The *user story* is that these adaptations should permit the generation of different simulated scenarios which can provide the analyst with possible answers to the following questions:

- What are the likely future impacts of the War in Ukraine, the Pandemic, and Brexit on the stratification of beef and dairy farms in Scotland between small, medium and large farms?
- Will there be any impact on the current phenomena of the "disappearing middle-sized farm"?
- How might policy instruments – i.e. i) GHG emissions-related CAP reform; ii) the easing of restrictions on visa applications for food supply chain workers; and iii) increased support for farms to import necessary inputs more cheaply and to access export markets – mitigate these impacts?
- What might be the impact on overall economic activity in the sector? Under what conditions might we witness a die-back of economic activity?
- What might be the collateral impacts on GHG emissions and biodiversity?

1. *Changes to the calculation of farm profits*
   1.1. **The inclusion of the price of labour**
      1.1.1. Description: The policy analyst has identified that the **Pandemic** and **Brexit** have caused increases in the price of labour and this factor is directly important for **profitability**
   1.2. **The inclusion of costs of importing/exporting to Europe**
      1.2.1. Description: The policy analyst has identified that **Brexit** has caused increases in the costs of importing/exporting from/to Europe and this factor is important for decreasing **profitability**
   1.3. **The inclusion of labour shortages**
      1.3.1. Description: The policy analyst has identified that the **Pandemic** and **Brexit** have caused increases in labour shortages and this factor is important for reducing **yield** and **labour needed for growing own feed.**
   1.4. **The inclusion of fossil fuel costs**
      1.4.1. Description: The policy analyst has identified that the **War in the Ukraine** has caused increases in fossil fuel costs that increase both the costs of **bought feed** and **own-grown feed**
   1.5. **The inclusion of changes in feed grain sources**
      1.5.1. Description: The policy analyst has identified that the **War in the Ukraine** has caused changes in grain sources that increase feed grain prices that increase the costs of **bought feed.**

2. *New scenario performance indicators*
   2.1. **Economic activity of farms in the supply chain**
      2.1.1. Description: The analyst would like the scenarios to produce aggregate data on the economic activity levels of farms in the supply chain, at constituency and national scale levels. He is interested in knowing when particular constituencies start to face a "die-back" in economic activity. The proxy offered by the analyst for this is the aggregate number per year of profit-seeking farms (including industrializing one) and diversifying farms that **generate a profit**.
   2.2. **Farm GHG emissions**
      2.2.1. Description: The analyst would like the scenarios to produce data on farm-generated **GHG emissions** at farm, constituency and national scale levels. The proxy offered by the analyst for this is the **number of cattle** on each farm.
   2.3. **Farm biodiversity levels**
      2.3.1. Description: The analyst would like the scenarios to produce data on on-farm biodiversity levels at farm, constituency and national scale levels. The proxy offered by the analyst for this is the **number of cattle** on each farm divided by the **area of land** of that farm.

Agile Requirements Specification Document – for Cow Policy Unit - 8th November 2022 – Author: matt.hare@hutton.ac.uk
4

1.4. The inclusion of fossil fuel costs

1.4.1. Description: The policy analyst has identified that the **War in the Ukraine** has caused increases in fossil fuel costs that increase both the costs of **bought feed** and **own-grown feed**

1.4.2. Design Spec: ***Task 1.4*** and ***Task 1.5*** require us to increase the cost of own-grown and bought feed. The cost of bought feed is given by **p-straw** and **p-hay**. Applying the approach proposed below for ***Task 3.3*** would enable adjustment of these prices without significant reworking of the code. From looking at the profit functions (**to-report calculate-profit-beef** and **to-report calculate-profit dairy**), it appears the cost of own-grown feed (**own-feed-straw** and **own-feed-grass**) is not currently considered as part of the profit calculations. It would therefore need to be added to the profit calculations. A straightforward approach might be to add sliders called **cost-own-straw** and **cost-own-grass** to the interface*. We would then add to the end of the line beginning **"set profit-beef …"** a snippet of code that deducts the own straw and own grass cost. To facilitate this, the total own-feed of each type used for beef and dairy cattle would need to be calculated first. *According to the John Nix Pocketbook for Farm Management (Redman, 2021 p.48), the yearly forage cost per all-year-round calving Friesian/Holstein dairy cow is **£122**. For a single lowland suckler cow, it is **£101** (p.61). For a single upland suckler cow, it is **£83** (p.63). For finishing of suckler bred store cattle, it is **£148** (p.67). We'll average the latter three values for beef cattle, and use the first value for dairy cattle. We'll then use these two numbers as our default slider values, allowing +/- of 50% from these defaults. Additional details of forage cost are provided on p.53 and p.91 of the John Nix Pocketbook. The numbers take account of seeds, fertilisers and sprays.

1.4.3. Coding: ***Task 1.4*** and ***Task 1.5*** Coded as per specification. I would ask the analyst to check my profit calculations in the `calculate-profit-beef` and `calculate-profit-diary` procedures.

Fig. 4. Top left: The first page of the requirements documents including the *user story*. **Top right**: requirements applying to the adaptation of the farm profit calculations and addition of new performance indicators. Numbers 1.x link to the nodes and arrows enumerated R1.x in Fig. 3. **Bottom**: The design specification (Section 1.4.2) added to requirement R1.4 by the modeller; and the programmer's notes on completing the coding adaptations (Section 1.4.3)

- Choosing the right machine on which to set up the code and run the legacy ABM was not straightforward. The legacy ABM used an old version of NetLogo, i.e., 6.1.1., with this version information contained in its code. However, initial attempts to run it on the programmer's Mac failed because even though version 6.1.1 could be installed, the `vid` extension (which was not needed) did not work because of lack of access to the correct Java versions that could, at the time, be installed on the Mac. Running the model on Linux, instead, worked.
- Demonstrating the replicability of the results of the legacy model will be a key part of verifying the adapted ABM. Unfortunately, the legacy ABM uses confidential spatial datasets for which the team would need to obtain new permissions, and users would require a decryption key before carrying out simulations.
- Time was required to understand how the original data used was wrangled - a process that requires easy access to original data processing scripts and metadata.
- No ODD of a complex ABM will provide an exhaustive overview of the coded programme it represents. There might therefore be small inconsistencies in them or missing descriptions of lower-level components, when compared with the code. It was necessary for the modeller to spend more time, therefore, to cross-check the ODD, the code and the causal loop model, to understand how specific parts of the legacy model were implemented so that design suggestions could be produced.
- Understanding the code was made harder and more time consuming by differences in coding styles between the programmer/modeller and the original ABM developer. For example, the latter saved results data as it was being produced, rather than at the end of simulation which was the preferred style of the programmer adapting the legacy ABM. There were also differences in naming conventions.
- Differences in such styles required the programmer to make a conscious decision as to whether they i) change their own style in favour of the style of the original programmer; ii) use their own style and have a potentially confusing mixture of programming styles in the adapted model, or iii) redo the legacy model to make it conform to their coding style, ensure coding consistency and improve ease by which future adaptations might be made.
- When it came to design specifications, the modeller noted that their choice of how to derive data would have a significant impact on the complexity or simplicity of the adaptations needed. An example of this is the specification of a user-input slider for deriving the `cost-of-own-straw` variable in R1.4 (see Fig. 5). Requiring this variable to be exogenous rather than calculated within the model using a new function, simplified the coding implementation. Strict, short time limits for adapting the model will thus tend to lead to simple, rather than more complex, adaptations. Also, unless we verify them with the policy maker and test the impacts of these decisions on the results generated by the adapted model, there is a risk that errors or model artefacts might creep in [10]. Naturally, this adds time to adaptation process.
- Linked to the above is the time-pressure-induced temptation to adopt a "quick and dirty" approach to design and re-coding of the adapted ABM, rather than a slower, more measured one. An example of this tension was identified by the team in the impact of the decision to split the modeller and programmer roles. This certainly slowed the process of model adaptation down and there was agreement that it could

have been done far quicker if the roles had been combined. However, it was also recognized in feedback discussions that, for example due to the complexity of the datasets being managed in the empirical ABM, such a split added another layer of quality control to the process by providing an opportunity for feedback (provided by the programmer) on the quality of the design specifications and for verification of the subsequent coding (provided by the modeller).

```
to-report calculate-profit-dairy
…
;; R1.4 adaptation -  Design spec: then add to the end of the line beginning "set
profit-dairy …" a snippet of code that deducts the own straw and own grass cost.
set profit-dairy (income-per-cow * cattle-dairy - p-straw * q-feed-straw - p-hay * q-
feed-grass
      - (cost-own-straw * own-feed-straw-per-cattle-dairy * cattle-dairy)
      - (cost-own-grass * own-feed-grass-per-cattle-dairy * cattle-dairy))
                             ;; new code – req R1.4
  ]
  report profit-dairy
end
```

```
@#$#@#$#@
GRAPHICS-WINDOW
….
;; R1.4 adaptation - design spec: add sliders called cost-own-straw and cost-own-grass
to the interface
SLIDER …
cost-own-straw
cost-own-straw
306 - 143
306 + 143
306.0
 …
HORIZONTAL
```

Fig. 5. Examples of the coding changes carried out for R1.4 *The inclusion of fossil fuel costs*. Top: changes to `set_profit_dairy`; Below: the required `cost-own-straw` GUI slider. n.b. code has been edited for presentational purposes.

4.1 Model Curation

The above issues suggest that if institutions are to be able to rapidly adapt legacy models to meet urgent policy queries *in a matter of weeks rather than months*, they will need to have a curated library of legacy models for responding to such queries. The nature of the barriers identified by our test even at such early stages of model adaptation - before phases of recalibration, verification, and validation - means there will probably not be enough time to respond, if such a library does not exist. Such a library will require an institution (or networks of institutions) to invest in model curation teams responsible for the curation of legacy models *in preparation* for possible future rapid adaptation requests from policy makers. As already proposed to support digital engineering enterprises, model curation involves "*lifecycle management, control, preservation and active enhancement of models and associated information to ensure value for current and future use, as well as repurposing beyond initial purpose and context*" which require professional curators ([2] p235). Such curators could ameliorate some of the issues covered in the section above by adopting the following roles. They would:

- identify high quality, legacy ABMs that have the potential to be of use in a specified range of policy areas.
- replicate the behaviour of the legacy ABM (rebuilding them if the identified legacy ABMs are not following the chosen coding style and documentation standards).
- ensure that the replicated legacy models and ancillary software versions are updated so that the legacy ABM can be run on each new generation of in-house machines.
- manage and update data licenses and permissions.
- continuously assimilate new data, ensuring the legacy ABM remains reflective of contemporary circumstances (e.g. extending "historical" datasets to the present day). To achieve this, sources of model data and data cleaning and formatting (data wrangling) procedures would have to be thoroughly documented.
- act as a trusted recipient and handler of confidential data needed for including sensitive information in policy ABM (a need identified in [5]).
- provide continuous professional development (CPD) training on the ABMs under curation to knowledge engineer/modeller/programmer teams.
- ensure all modelling teams are employing in-house coding style and documentation standards for current and future ABM development (including code layout, naming conventions, code annotation and metadata recording).
- verify in-house ABM ODDs, backed up by formal ontological descriptions.
- maintain institutional memory of in-house ABM should the developers leave the organisation.
- promote, within the ABM community, requirements for legacy models if their developers want them to be used to support rapid response model adaptation using our protocol. For example, requirements would include the provision by the ABM developer of ODD documentation that has been verified against the model code, as well as clear documentation of code and of any data wrangling. Since the causal loop model of RISC provided our modeller (responsible for the design specification) extra clarity about the dynamics of this ABM, the provision of such a representation alongside the ODD might also be considered a useful requirement.
- curate reusable modules, such as those being considered by the 'Reusable Building Blocks' community[3], to allow the composition of ABMs to fit policy queries for which there are no suitable legacy ABMs.

Model curation activities would require significant financial resources to cover work that would not normally be included in research grants. A possible solution would be to seek investment under "underpinning capacity" funding to boost the type of trusted, rapid, policy-responsive large scale modelling called for by Squazzoni et al. [5].

Acknowledgements. This work was supported by the Scottish Government Rural and Environmental Science and Analytical Services Division (RESAS) (project reference JHI-C5-1). We are also grateful for the work of the two reviewers of this paper.

[3] https://bit.ly/RBB_template ; https://bit.ly/RBB_forum

References

1. Borowski, I. and M.P. Hare, *Exploring the Gap Between Water Managers and Researchers: Difficulties of Model-Based Tools to Support Practical Water Management.* Water Resources Management. **21**: p. 1049-1074 (2007).
2. Rhodes, D.H., *Model Curation: Requisite Leadership and Practice in Digital Engineering Enterprises.* Procedia Computer Science. **153**: p. 233-241 (2019).
3. Polhill, J.G., et al., *Crossing the chasm: a 'tube-map' for agent-based social simulation of policy scenarios in spatially-distributed systems.* GeoInformatica. **23**(2): p. 169-199 (2019).
4. Hedelin, B., et al., *What's left before participatory modeling can fully support real-world environmental planning processes: A case study review.* Environmental Modelling & Software. **143**: p. 105073 (2021).
5. Squazzoni, F., et al., *Computational Models That Matter During a Global Pandemic Outbreak: A Call to Action.* Journal of Artificial Societies and Social Simulation. **23**(2): p. 10 (2020).
6. Thomas, A., et al., *Rapid adaptive modelling for policy support towards achieving Sustainable Development Goals: Brexit and the livestock sector in Wales.* Environmental Science & Policy. **125**: p. 21-31 (2021).
7. Meijer, K., et al., *Fit for purpose? Rapid development of water allocation models using global data: Application for the Upper Niger Basin.* Environmental Modelling & Software. **145**: p. 105168 (2021).
8. Manzoor, U. and B. Zafar, *Multi-Agent Modeling Toolkit – MAMT.* Simulation Modelling Practice and Theory. **49**: p. 215-227 (2014).
9. Roxburgh, S.H. and I.D. Davies, *COINS: an integrative modelling shell for carbon accounting and general ecological analysis.* Environmental Modelling & Software. **21**(3): p. 359-374 (2006).
10. Galan, J.M., et al., *Errors and Artefacts in Agent-Based Modellin.* Journal of Artificial Societies and Social Simulation. **12**(1): p. 1 (2009).
11. Castro, A.G., et al., *The use of concept maps during knowledge elicitation in ontology development processes – the nutrigenomics use case.* BMC Bioinformatics. **7**(1): p. 267 (2006).
12. Grimm, V., et al., *The ODD Protocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism.* Journal of Artificial Societies and Social Simulation. **23**(2): p. 7 (2020).
13. Vennix, J.A.M., *Group Model Building*. Chichester, New York: J. Wiley (1996).
14. McGeorge, P. and G. Rugg, *The uses of 'contrived' knowledge elicitation techniques.* Expert Systems. **9**(3): p. 149-154 (1992).
15. Ge, J., et al., *Not one Brexit: How local context and social processes influence policy analysis.* PLOS ONE. **13**(12): p. e0208451 (2018).
16. Tufts University. *VUE User Guide*. Available from: https://sites.tufts.edu/vue/ (2023).
17. Stellman, A. and J. Greene, *Learning Agile: Understanding Scrum, XP, Lean and Kanban*. Sebastopol, CA, USA: O'Reilly (2015).