## Thesis Corrections — Response Doc

## Tom Wallis fao Steve Marsh, Phil Trinder

REPORT

The thesis represents a considerable amount of research software engineering and some experimentation that explores the potential of aspect orientation for developing and adapting simulations. There are, however, such substantial issues with the presentation of the work that the key research contributions are not made clear.

The thesis is quite hard to understand, and many aspects became clearer in the Viva. Specifically concepts are used before/without definition, and even when defined there are no examples to illustrate. Design and methodological choices are made with justification, or discussion of alternatives.

The research contribution of Chapters 4, 5 and 6 are unclear in the thesis text. While they clearly represent a substantial software engineering effort, they do not adequately explain why they are research software engineering. The research vision and the contributions of these Chapters were clarified in the Viva.

In Chapter 7 a careful experimental methodology is described, although alternative techniques are not adequately discussed. There are clearly articulated research questions, and some limited but credible experimentation, that goes some way towards establishing them. However the claims need to be phrased carefully to reflect the scale of the study.

MAJOR ISSUES

M1. The thesis needs to be in correct UoG format.

As discussed following the viva, the format of the thesis already adheres to university guidelines. It also appears to be acceptable on the basis that other theses have been submitted using the same class file specifying the layout. As a result, the format of the thesis has not been changed.

M2. The research contribution of Chapters 4, 5 and 6 should be made clear. The current titles suggest that they are simply software engineering tasks, and not RESEARCH software engineering: recommend changing them. Some specific elements are elaborated below, and in the annotated thesis copies.

The title of Chapter 4 has been changed to “A Framework for Aspect-Oriented Modelling”. The title of Chapter 5 was changed to “RPGLite” to avoid biasing readers’ perspectives on the chapter’s contributions, and the chapter’s contributions were summarised more explicitly in a new first paragraph (page 141) as well as in a new introductory subsection (Section 5.1.1, page 143), and its Summary subsection (Section 5.4.1, page 163). The title of Chapter 6 was changed to “Modelling with Aspect-Oriented Programming“ and the chapter’s contributions were also clarified in Section 6.1.1 (page 167) and again in Section 6.7 (page 205).

M3. It is crucial when making a design or methodological choice to discuss alternatives, and hence justify your choice. Some specific examples are identified in the annotated thesis copies.

Several instances of missing justifications for decisions and chosen methodologies were rectified. For example, alternatives to import hook weaving are more clearly described as such in Section 4.3. Additional explanatory text was provided at the start of Section 4.3 (page 99) and Section 4.4 (page 105) to clarify that Sections 4.3.1 (page 100) and 4.3.2 (page 103) both contain potential hook weaving techniques, and that import hook weaving is chosen as opposed to the other alternatives discussed. The introduction to Section 4.3 (page 99) qualifies what is being discussed (techniques to weave hooks using only native Python) and why those criteria are appropriate. Section 4.4’s introductory text (page 99) now reiterates the reasons as to why import hook weaving is preferable to the alternatives discussed in Section 4.3, and makes clear that Section 4.4 describes the implementation of an aspect-oriented programming framework given that choice.

M4. The experiments in Chapter 7 are of limited scale, e.g. experiments on a single model (RQ1), and on two closely-related versions of that model (RQ3). Hence it is essential to express the claims of how thoroughly the RQs are answered far more carefully.

When discussing the results of (and contributions resulting from) the experiments described in Chapter 7, their limitation stemming from the scope of the study is explicitly discussed. Modifications were made in both Chapter 7 and in Chapter 9, where those contributions are summarised again. For examples of these changes see Section 7.2.3 (page 218), Section 7.4.3 (page 228), and section 9.2 (page 262).

M5 For clarity and precision recommend giving different names to the PyDySoFu software described in Chapters 3 and 4. This means you can avoid

+ cumbersome references like “PyDySoFu’s updated implementation”, or “this implementation of PyDySoFu”, and

+ avoid ambiguity, so make it clear when you write “PyDySoFu” which version you intend.

The old version of the tool is now referred to as PyDySoFu, and its new implementation is named PDSF3 (as it is Python3-compatible, and “PDSF” is easier to read than “PyDySoFu”. To ensure all references to the tool were updated (so that phrases around the tool such as “updated implementation” could be removed or changed), the LaTeX macro used to refer to the tool was removed. Any un-checked uses of the tool’s name would therefore cause the document to fail to compile. Two new macros were introduced to refer to PyDySoFu and PDSF3 specifically, and all uses of the original macro were replaced with one or the other until the document compiled. The text around each use of the original macro was checked after every edit to ensure no cumbersome references and ambiguity remained. Effectively all qualifications as to which tool was being discussed could be removed (except where introducing the terms), and the name change reduced ambiguity.

M6a Section 4.3 is quite intricate, and is explained by around 20 pages of text referencing some useful examples. Strongly recommend that you add concise semi-formal descriptions of key elements. For example provide a pseudo-code algorithm, or UML Activity Diagram, for the PyDySoFu "with ... import" process.

A pseudocode representation of the most complex code snippet in the section can now be found in Section 4.4.5 (page 118). Similar pseudocode snippets are repeated where useful to explain more complicated concepts, as the reader is likely to benefit from a second point of reference after several pages of the more dense sections in the thesis; for example, pseudocode is now provided in the explanation of the application of fuzzers in section 4.4.5 (page 120) and for the construction of around advice in Section 4.4.5 (page 123). To clarify the construction of a wrapper which contains all around advice and the target of aspect application, a flowchart is added on page 120. Section 4.4.1 now includes a diagram to better explain the process of aspect hook weaving (page 107), to complement the enumerated list it appears alongside, which was also rewritten for clarity. This is to better prepare the reader to understand the section which this correction concerns, as that section describes the implementation of the process which the new figure illustrates.

M6b If Import Hook Weaving is a key technical contribution it is ESSENTIAL to establish it’s novelty. This entails carefully explaining how it differs from weaving in AspectJ & other Aspect-oriented languages (M2).

The novelty of import hook weaving is now discussed in Section 4.4.7 (page 127), which compares PDSF3 to other frameworks, in particular with regards their implementations of aspect hooks. The novelty of import hooks weavings specifically discussed and established.

M7 Section 4.4 it’s unclear which of the optimisations, e.g. deep weaving and static weaving, are established concepts. If they have previously been implemented cite prior work (GC1). If not emphasise that it's new and you are defining it. Careful discussion is required to establish the novelty of the chapter (M2).

Each subsection describing an optimisation within Section 4.5 (previously Section 4.4) has been given a final paragraph at the end of the subsection, which describes whether the optimisation is novel and compares it to other aspect-oriented programming frameworks as appropriate. Also, examples of other frameworks’ implementations are illustrated through figures containing code snippets relevant to the feature in the compared frameworks where appropriate. Examples of these improvements are the discussion of ordering aspect application in Section 4.5.3 (page 134) and cached fuzzing in Section 4.5.4 (page 135).

M8 Discuss what will motivate people to build “Aspect-Oriented Models” (Chapt 6)?

Several motivations to build aspect-oriented models are discussed in a new subsection, Section 1.2.4 (page 29). The changes were added in the Introduction chapter to address other comments within the annotated .pdf (which requested that examples be given to justify aspect-oriented modelling frameworks requiring new types of join-points), and to give the reader early context for the contributions in the thesis. Adding the motivations early in the thesis (i.e. introduction rather than Chapter 6) also provides motivating context for building frameworks for aspect-oriented modelling (Chapter 4).

GENERAL COMMENTS

To be addressed throughout the thesis. Only some examples are noted.

GC1 Define concepts before use, e.g. “obliviousness” is used on page 17, and only defined on page 22. You often need to

+ give a precise name to the concept (and use it consistently thereafter)

+ cite a definition, and/or

+ give examples to illustrate (GC3)

Definition of “obliviousness” is also given on page 21 (which was page 17 before corrections were made), but was not clearly signposted in the way other definitions on the page were (e.g. surrounding the defined term in quotation marks). Definition of “obliviousness” was reworked on page 21 (prev. 17) to be clearer and also better signposted to readers. Other definitions were moved and rewritten for clarity and to ensure concept is defined before use, e.g. definitions of “model”, “simulation”, and “socio-technical system” were moved to Sections 1.2.1 (socio-technical system, page 25) and 1.2.2 (model & simulation, page 26) to precede motivations of aspect-oriented modelling, which were moved to Section 1.2.4 to avoid use of terms before definition. Additional citations were also given, such as when describing Python’s import cache in Section 4.4.3 (page 112). The definition of monkey-patching in Section 4.3.1 (page 100) is rewritten for clarity and precision, given a citation, and is given a Python code example to illustrate the concept. In other scenarios it appeared that a new term was being used without definition due to missing words in a sentence; for example, in Section 4.4.2 (page 103), the term “prospective imports” was noted as an undefined term in the annotated .pdf sent after the viva; however, the sentence should have read “[…] only when modules containing prospective join-points are imported” — cases such as these were corrected as found in both the annotated .pdf and as found in a search of the thesis performed before the viva. To avoid using confusing phrasing, the example above was further edited to replace “prospective” with “potential” to keep phrasing consistent with the previous subsection.

GC2 Sentences must be grammatical! Maybe use an AI to detect?

Grammar mistakes included in both the annotated .pdf and found in my own annotated thesis copy (made in preparation for the viva) were corrected by editing the entire thesis end-to-end. These changes exist throughout the thesis. For examples, see the first sentence of Section 1.2 (page 25) which was rephrased to be grammatically correct. Also see the first paragraph of the static weaving optimisation described in Section 4.5.2 (page 133) which was entirely rewritten for clarity and correctness.

GC3 When describing concepts give examples to illustrate what you mean. Moreover when defining related concepts, e.g. models and simulations, use matching examples.

In Section 1.2.4 (page 30), examples of specific use-cases are identified to clarify the point made and a cross-reference to a section detailing that use-case is also included to further clarify. Section 1.2.2 (page 26) includes examples of both models and simulations to clarify the definitions given. Section 2.4.1 (page 75) includes examples to differentiate the concepts of “faithful” and “accurate” models. Additional diagrams were also included to clarify some concepts — for example, Section 4.4.1 (page 107) is updated to include a UML activity diagram summarising a high-level description of import hook weaving. Another example can be found in Section 4.4.3 (page 112) which now contains an example of a Python project with several modules to illustrate how PDSF3’s manipulation of Python’s import cache affects modules which do not include any uses of PDSF3’s AspectHooks.

GC4 Add citations to evidence key assertions

Citations were added wherever it was found that they were missing — for example, in Section 1.2.5 (page 32) where discussing the legibility of aspect-oriented programs. The annotated .pdf also noted some key assertions which arise from the nature of aspect-oriented programming but were not clearly and sufficiently explained; Section 1.2.5 (page 32) also contains an example of this, regarding the fact that aspect-oriented modelling requires a framework which supports “within”-style join-points. This point was rewritten to be more clearly explained, and references to earlier subsections are included to further explain the point made. Care was taken to ensure that key assertions which were included when making corrections were not missing citations. For example, see the discussion of cross-validation techniques in Sections 7.1.1 (page 208) and 7.1.2 (page 210).

GC5 Justify the choices you make in your experiment designs, and discuss alternatives.

Justifications for decisions made in implementing tools and designing experiments have been added where missing, and alternatives are now also discussed when missing. Most significantly, Section 7.1 was largely rewritten to more clearly and precisely describe experimental methodology. It includes two new subsections: Section 7.1.1 (page 208) which explains model overfitting and identifies cross-validation as an appropriate technique to identify overfitting when selecting model parameters, and Section 7.1.2 (page 210), which describes several cross-validation techniques and identifies Leave-One-Out Cross-Validation (LOOCV) as the preferred type of cross-validation for the experiments presented in Chapter 7. Section 7.1.2 also describes more precisely how LOOCV works as well as related cross-validation approaches. Rationale supporting parameter selection was clarified in Section Section 7.3.2 (page . Some justifications and discussions of alternative choices for the design of experiments were also updated which were unclearly written. For example, in Section 6.4.4 (page 181) the Birch curve was not clearly proposed in comparison to other curves which had been surveyed earlier in the chapter; the section has been updated to correct this. In Section 4.3 (page 99) alternatives to import hook weaving were given but not clearly linked to import hook weaving itself. Justifications for seeking native-Python implementations (as opposed to DSL-based or runtime-based implementations) are now provided at the start of Section 4.3 (page 99). This is summarised again for clarity at the beginning of Section 4.4 (page 105), explicitly noting that import hook weaving is an alternative to the others surveyed.

GC5b Specifically explain how you chose the sets of numeric parameter values

An explanation of the method used when selecting sets of possible parameters is now given in Section 7.3.2 (page 221), along with some alternative approaches for parameter selection which future studies could adopt.

GC6 Caption, reference and discuss tables as Tables, and not Figures.

Changed enclosing environment of tabular latex blocks to tables instead of figures, fixing how they are displayed. As they are consistently referred to through cleverref’s \cref{} macro, they are now referenced and discussed as tables instead of figures automatically also. For example, see the tables in Section 7.2.2 (page 221).

GC7 In the Software Architecture diagrams in Chapter 7 (Figs 7.1, 7.4 etc.) state the diagramming notation used in the caption & describing text

The diagrams are now described as UML activity diagrams, with modifications to describe the application of aspects (dashed boxes and lines) explained using a reference to the business process metamodelling language which inspired the notation, AOPML. This change was also made to similar diagrams in Section 6.5.1 (page 185). Similar changed diagrams can be found in Sections 7.2.1 (page 216) and 7.3.1 (page 220).

GC8 In the Software Architecture diagrams in Chapter 7 (Figs 7.1, 7.4 etc.) emphasise the new elements for this experiment, e.g. use colour, emboldening, whatever…

New elements for each model are now coloured red and green for the models of learning and selection from prior distribution respectively. This is also explained in the caption for the figures, and the change was also made to Figure 6.3 in Section 6.5.1 (page 185) where a similar diagram was. Examples of changed figures in Chapter 7 can be found in Section 7.2.1 (page 216) and Section 7.3.1 (page 220).

GC9 Give Section cross-references rather than Chapter cross-references

The thesis was edited to include more specific cross-references wherever appropriate, such as in the first paragraph of Section 1.2 (page 25) or the first paragraph of Section 4.3 (page 99).

TYPOS/Minor

There are many comments in the PDF versions of the thesis shared with you.

This .pdf was used to edit the entire thesis, addressing each comment as much as possible.

Believe that Section and Chapter are conventionally capitalized.

Cleverref package was given the option “capitalize”, which changed this throughout the document.

INTRODUCTION: Provide better evidence of the challenges posed by evolving research software.

Challenges posed by evolving research software were better explained in Section 1.1 (page 23) and a citation was added. A rewrite of the motivations for using aspect-oriented modelling further clarifies this and adds more citations, and can be found in Section 1.2.4 (page 29).

Birch is a proper noun, and so Birch curve should be capitalized in the text.

Document source was searched for lowercase “birch”, and instances were replaced with “Birch”.

The references need to be improved to be correct and complete. Some infelicities noted in the PDF.

It was noted in the annotated .pdf sent with the corrections that some references had strange IDs — the bibliography style has been change to numeric instead of alphabetical to correct. URLs included in references which failed to wrap to fit page margins are corrected using the xurl package. Names of authors on GitHub repositories are reformatted to avoid “contributors” being interpreted by latex as a lead author name. Ambiguity around my own publications being published as both Tom Wallis and William Wallis were clarified by adding notes to relevant bibliography entries. Missing and incorrectly formatted fields were identified and corrected, such as the string of missing characters in Shugurov & Mitsyuk citation (number 132, page 281) or missing journal names entries such as Lock, Storer & Sommerville (number 94, page 277).

Is “accurate” in RQ1 and RQ2 the same? If so, does RQ2 need to be reworded. If not do you need another concept of “faithful” representation in RQ2?

As discussed in the viva, RQ1 and RQ2 refer to subtly different things, and “accurate” is used in an ambiguous manner across the two questions. RQ2 was reworded to use the term “faithfully” instead of “accurately”, and the two terms are defined in the discussion of research questions found in Section 2.4.1 (page 75).The definitions are accompanied by examples, as requested in GC3.