

Report on the development of the Toolkit

Y. Andrianakis and P.C. Challenor

March 23, 2009

1 Introduction

The Toolkit is one of the primary means for disseminating the research that takes place in the MUCM project. Its aim is to address people who work with numerical models and could benefit from the research output of MUCM, but at the same time, it is aimed to researchers that work in the field of statistical model emulation. The first group is targeted mainly via pages containing the core science behind model emulation. These pages provide a detailed procedural description for implementing well established methods that have either been developed in the MUCM project, or have been used within its scope. The researchers/scientists can benefit from pages that contain the latest developments on the research that is being pursued in MUCM, which can either have matured into an application-ready tool or can still be at a development stage.

This document attempts to describe the current state of the Toolkit, to highlight some of its current limitations and to propose directions that its development could follow in an effort to overcome them.

2 Organisation of the Toolkit

Every page in the Toolkit belongs in one of the categories listed in table 1. The table also gives a description of the purpose of each page, which should be to some extent obvious from its name. More information on the function of each page in the Toolkit can be found in page [MetaToolkitStyleGuide]. Although the above taxonomy seems sufficient in describing the different types of pages, it does not provide a method for organising the material that is found in the Toolkit according to a subject, nor does it define relations between different types of pages on the same subject.

Page name	Prefix	Description
Procedure	Proc	Procedural description of an operation
Discussion	Disc	Discussion of issues or possibly optional details
Alternatives	Alt	How to choose between alternatives
Definition	Def	A definition of a term or a concept
Thread	Thread	An entry-point into the Toolkit; a narrative using Toolkit pages
Subject	Subj	A subject area; a set of pages related by topic
Meta	Meta	Any other page, not part of the Toolkit, but still in the Wiki
Template	Template	Suggested templates for the other pages

Table 1: Page Types

A first attempt towards a content based organisation of the Toolkit was the introduction of *Subject Areas*. The Subject Areas were meant to organise the contents by topic, with each containing a particular segment or theme of the Toolkit. Although Subject Areas had the potential of grouping pages according to their content, they lacked a mechanism that determined the connections that exist between different pages on the same theme. For example, an ‘Alternatives’ page on design options and a ‘Procedure’ page on Latin Hypercube design, would both go in a ‘Design’ Subject Area, but there would be no provision for acknowledging that the Latin Hypercube page implements one of the methods discussed in the ‘Alternatives’ page.

Failing to define the connections between pages on the same theme, had two main drawbacks, one affecting the users of the Toolkit and another one affecting its authors. The users of the Toolkit, especially the less experienced ones, were not given enough guidance as to how to complete a specific task and were only provided with a collection of related pages. Ideally, the pages within a Subject Area should be linked by a thread that the user can follow and accomplish all or part of the theme that the Subject Area is treating. Regarding the contributors, the lack of a specific structure did not help defining which pages needed authoring and uploading, so as the Subject Area to be complete. Taking into account that writing a page is a time consuming process, the lack of a clear incentive hindered the rate by which new pages were written.

The introduction of *Threads* tries to remedy this lack of connectivity between pages with related content and tackle the above two problems. While a Thread typically contains pages which are related by a common underlying theme, its most important characteristic is that it brings together pages with the purpose of achieving a specific and well defined goal. Hence, apart from containing pages related by a common theme, a Thread also provides a natural ordering for the pages with the purpose of guiding the user to accomplish a particular MUCM application.

An example of such a Thread, which also aims to serve as a prototype for others, is the [Thread-CoreGP]. This Thread treats the core problem, which concerns a single simulator with one deterministic output and uses the fully Bayesian approach for modelling (see the Thread page for a full specification of the core problem). Using a narrative approach, this Thread defines the GP model, the prior distributions and basic methods for designing an experiment. Then, it analytically explains how the emulator can be fitted to a series of observations and finally describes a number of common tasks that can be performed once the emulator is designed, such as prediction, uncertainty and sensitivity analysis. In its development, a Thread can contain Definitions, provide Alternatives, give implementation Procedures, and hold Discussions on issues that may arise during implementation or when particular choices are being made. In other words, a Thread can provide a framework that incorporates all the types of pages contained in the Toolkit, but in a structured and goal oriented fashion.

Currently, there is provision for two types of Threads. Firstly, there are the Main Threads, which cover basic and well established methods. The Main Threads aim to take the user in detailed steps through the entire the process of implementing a MUCM application. The second type consists of Technical Threads. These could be of more interest to researchers in the area and discuss technology that is still under development and has not yet matured into an application.

The Threads are hoped to serve a two-fold purpose: the first is to provide a structured entry point to the Toolkit for users who want to use a particular MUCM application. Secondly, their structured nature should allow the creation of an almost complete list of pages for a Thread, before setting out to write the actual pages. In this way, the progress of the Thread can be quantified, the gaps in its narrative can be identified easier and ultimately this should provide a clearer focus on which pages the authors should contribute for the Thread to be complete. It was acknowledged that a shortcoming of the Subject Areas was that they did not provide the authors with enough guidance as to which pages they needed to contribute and the notion of uploading *all* pages related to a subject was rather vague. The more structured and application oriented nature of Threads is

hoped to overcome this deficit.

At the current stage, the ThreadCoreGP is almost completed and will form the main part of the Toolkit's first version that will go public. There is also a twin Thread that treats the core problem as well, but under a Bayes Linear point of view. This Thread is also under development. Apart from the two Threads treating the core problem, there are also plans on developing Threads for more specific cases. For example, Threads could be developed for the case where we are interested in more than one outputs, when we have more than one simulators, when we also observe derivatives of the outputs and so on. It is hoped that the paradigm of the core Thread will facilitate the development of the subsequent Threads by serving as a model on which they can be based on.

The adoption of Threads as the main method for a content based organisation of the Toolkit, has made the need for the subject areas obsolete. Therefore, the subject areas have been removed from the pages of the ThreadCoreGP, and will gradually be removed from the rest of the pages as well. There is however a thought of introducing keywords to each page, so as to facilitate a topic based search of the toolkit. The implementation details of the keywords scheme are not yet clear. A possible way would be with the introduction of metadata to the pages. These details are to be specified in due course.

3 Example pages

In the midterm meeting, it transpired that the addition of Examples to each Thread would be a very welcome enhancement to the Toolkit. These could come in the form of pages that contain initially elementary but maybe later more advanced Examples of the application that is covered in each Thread. Ultimately, Case Studies can be seen as examples, but because of their size and complexity we will not consider them as such. The Toolkit contains a significant portion of non elementary maths and the material can be daunting to new users, straying them away from using methods that could eventually be beneficial to their work. It is hoped that the introduction of simple Examples, which are however related to real world scenarios, will help, especially the less experienced users, to comprehend the presented material and appreciate how the methods could be relevant to their own problems.

We think that the Example pages will mainly benefit users who are new to the MUCM methods. For this reason, it is important that we create solid, working Examples for the core Threads of the Toolkit, that is, the ThreadCoreGP and the ThreadCoreBL. It is anticipated that a user who has implemented the core methods of the project will need less guidance when implementing more advanced techniques. Nevertheless, Example pages for the latter can be included, especially if these are rather involved or have delicate implementation points.

A new page type (e.g. Example pages) could be created so as to accommodate the Examples. The Examples could run in parallel with a Thread, following closely its development and helping the user to consolidate the theoretical concepts presented there. It could also be possible that more than one Examples (e.g. one elementary and one more advanced) are given for each Thread. A link to the Example that accompanies every Thread could be given at the end of its main page. However, links to the individual pages of each Example could be located at the end of the respective Procedure pages as well. Finally, the code used for generating each Example could also be provided when possible. This could help disambiguate even the smallest details of the implementation and could serve as a basis for the users to start developing their own applications.

4 Implementation pages

The implementation of some MUCM methods can require a number of details that do not necessarily fit in a Procedure page. Some methods for example, can create numerical instabilities, and guidance must be given to the users as to how to sidestep them. If this type of guidance is more substantial than a tip that could fit in a Procedure page without interrupting its flow, a new type of page - the Implementation page - can be introduced. This page type can then hold all the implementation details of the procedure. Links to these pages are expected to be found within the related Procedure pages. As a prefix for these pages we propose the wiki word 'Implement'.

5 Introductory pages

Except for the technical contents, it would be also useful to have one or more Introductory pages to the Toolkit. Apart from acting as welcome pages for the new users, these pages can introduce the Toolkit and maybe set the bigger picture before delving in the often intricate details that follow. Additionally, some of these pages could also act as a road map pointing the users to the Threads that would be more suitable for tackling their problems.

6 Software

A decision taken at the designing stage of the Toolkit, was that the methods it contains would not be written in a particular programming language, but they should rather be presented in the form of a generic recipe book. This decision was taken because a commitment to a specific programming language would exclude or discourage users coming from different computing backgrounds, from adopting methods developed in MUCM.

While we stand by this decision, we still find useful to provide users with code or software that has either been written by MUCM members, or provided freely by other researchers. This could be as diverse as a segment of code that implements a particular procedure, or a complete software package such as GEM-SA. We also support distributing code written in a variety of programming languages, so as to increase the number of users that can benefit from it.

Although we encourage providing software that implements methods described in the Toolkit, there is no intention of integrating this software in the Toolkit itself. For this reason, software will be offered either by links to uploaded code or executable files, or by links to external websites that host larger software packages. There will not be however a provision for dedicated software pages within the Toolkit. For example, should a researcher want to make public a number of functions that implement a specific procedure along with comments on their use, the best practice would be to host these in a personal web page and link from a relevant Toolkit page.

Finally, any page linking to software should also make clear that the software does not constitute a part of the Toolkit and is provided 'as is', with no guarantees as to its functionality. The conditions under which the software is provided are detailed in a disclaimer meta-page [MetaSoftwareDisclaimer], that should be placed next to each software link.

7 Notation

Maintaining a uniform notation across the toolkit will improve its usability and raise its quality standard, but it is also a very challenging task. Ideally, we would like to have a common and one to one notation between symbols and variables across the toolkit. Considering however the sheer volume of parameters involved across the spectrum of MUCM applications, one realises that this is a difficult goal to attain.

On the other hand, at the current stage a small part of the Toolkit is only written, which offers the opportunity of standardising a notation and adhering to it when building the following parts. Although a complete notation table could probably not be built at this stage, we believe that we should fix the notation for the most elementary quantities, that are used across most, if not all, the areas of MUCM. We think that this notation to a large extent coincides with the notation that is currently used in ThreadCoreGP (although some parts of it are likely to change), but should also extend to other quantities that are not covered in ThreadCoreGP but are considered fundamental e.g. multiple outputs, real world observations etc.

In an effort to standardise the used notation, Yiannis Andrianakis will set up a page that contains the symbols used for the quantities that are common across the different areas of MUCM, and circulate it to the team members. We then hope to have a discussion and jointly decide on a ‘basic’ notation that will be acceptable from all the team members, and that is less likely to create conflicts when new areas of the project are added in the toolkit.

Once the basic notation page is set, the authors will be expected to adhere to it when authoring new pages. Additionally, when a new (perhaps major) quantity is introduced, its symbol should be uploaded in the list and the team can be notified about the new entry.

As the development of the Toolkit progresses, we could also build a nomenclature table for each Thread or, perhaps more ambitiously, for the entire Toolkit. Such tables should be very useful to the new users who are not familiar with the plethora of terms that populate the model emulation literature, but will also be a handy reference for the more experienced users.

8 Comments

Once the Wiki is public, it would be useful to have a system by which the users could post comments on the various pages. This would be a tremendous source of feedback that should improve the material of the Toolkit and its presentation.

9 Graph tool

In the midterm meeting there were discussions regarding the development of a tool that would automatically generate a graph that contains all the pages that are linked from a particular page of the Toolkit. Such a system would provide a very useful pictorial representation of a Thread and assist its planning and development. Also, such a graph could be placed at the main page of each Thread, acting as a roadmap to its contents.

10 Intellectual property issues

Apart from well established methods, the Toolkit also aims to contain the latest research that is pursued in MUCM. Two issues could arise here. The first regards safeguarding the teams intellectual property. If the material is about to be published we have to ensure that this does not leak from the Toolkit before it reaches the publishing editor. On the other hand, we have to make sure that when copyrighted material is written for the Toolkit, its uploading does not violate any existing copyright agreements.

11 Governance

Tony O'Hagan has kindly agreed to be the Toolkit's overall editor. We think that this is very fortunate for the Toolkit's development, as someone with the experience and the weight of opinion of Tony would be the ideal person for the job. Yiannis Andrianakis can be the assistant editor, responsible for the day to day maintenance of the Toolkit.

In a similar way that each Subject Area had an editor, each Thread should have an editor as well. The Thread editor should be the person designing the Thread's overall structure and determining which parts of MUCM research this will include. The exact contents of a Thread can obviously be discussed with the other team members and the Toolkit's overall editor, but it is ultimately the Thread editor that makes decisions and assumes responsibility of the Thread's development. The Thread editors can also assign the persons they think more appropriate for writing up of the different parts of a Thread. We see that the job of a Thread editor is best suited to a PI.

It would also be advantageous if the PIs wrote the core pages of each Thread, that is, those pages that do not contain the technical details, but rather set the context on which the more technical pages are then attached. Again, someone with longer experience in the field can present the more general aspects of a subject in a more coherent and educational manner. The other team members could be better suited for writing up the more technical pages. It is very often that the RA-PS are those who implement the methods of the project, so they have the necessary experience for writing up for example, Procedure, Alternatives or Discussion pages. Also, writing up a technical page can be an excellent method for the less experienced members of the team to reinforce their knowledge.

Many of the technical pages will include a high mathematical content, which will need to be proofread before it is made public. Additionally, less technical pages need to be checked for the accuracy of their contents as well as for their coherency with other pages, typographical errors and so on. This creates the need for a system by which one or more persons proofread each page, before this goes to the public domain.

An idea could be that the Toolkit editor checks every page before it is made public. That would require however a tremendous effort from the side of the editor. An alternative would be that the Thread editor checks each of the Thread's pages. Currently, the CoreGP Thread numbers approximately 70 pages and is likely to grow even further; so proofreading a whole Thread is still a very demanding task. This is especially true when we consider that a large number of pages contain lengthy equations which should be checked thoroughly before made public. At the same time we would like to have each page checked by a person who has the appropriate knowledge for doing so.

A signing off system that would fulfill the above requirements could be one according to which, the author of each page assigns the person who thinks more appropriate for proofreading his writings. In this way the proofreading system is decentralised, as the entire MUCM team will participate in both the writing and the proofreading processes. Additionally, we can secure that the pages

are proofread by people who are well equipped for doing so, since each author should know who in the team has the necessary skills for commenting on his scripts. An issue that needs to be addressed here, is that some team members do not get excessive proofreading or writing workload. The division of the project into discrete themes could safeguard this to some extent, while there is also trust in the mutual understanding that exists between the team members that the workload will be spread evenly. Finally, a provision should be made in the Wiki software for enabling the person who has proofread a page to sign off that he has actually done so.

The Toolkit's pages can be edited from any member of the MUCM team. However, we ask that the author is notified when changes are made to a page that he/she has authored. The author will then have to approve the changes and notify in turn the reviewer of the page, who will have to sign it off. We see this procedure as important in maintaining the quality of the Toolkit, as it ensures that a page is checked by at least two specialists before it is made public.

12 Release cycle

The first part of the toolkit will go live at the end of March 2009. This will consist of the ThreadCoreGP and most parts of ThreadCoreBL. With the MUCM project finishing in October 2010, this leaves one year and a half for the completion of the Toolkit. We propose a release cycle of three months. This will allow six more releases to be made until the end of the project. Additionally, the exact dates of the releases could be scheduled in such a way so that they coincide with the three month board meetings that are currently in place.

Given that the Toolkit is a major deliverable of the project, in order to help planning and to ensure that the Toolkit will be delivered by October 2010, we propose that each release forms a milestone for each of the work plans. This should help spreading the workload evenly in the next year and a half, thus minimising the disruption to the progress of research, while ensuring at the same time the timely delivery of the Toolkit.

The frequent releases of the toolkit will create a need for a versioning system. Each webpage should mention clearly which version of the toolkit is the user currently working with. Between releases, it will not be possible to modify the online version of the toolkit, with the exception of correcting errors that someone might come across. For this reason, editing permission for the live version will be granted only to a small number of people (possibly the assistant editor and the editor) and all modification requests should be channeled through them.

13 Conclusion

With the first live version of the Toolkit being imminent, a number of changes are being proposed and implemented. A major change in the Toolkit's structure is the replacement of the Subject Areas with Threads. Being more application or task oriented, the Threads are hoped to aid the further development of the Toolkit and act as a structured entry point to users interested in a particular MUCM application.

A proposed addition to the Toolkit are the Example pages; these will contain elementary or more advanced examples, that will accompany a Thread, following step by step its development. The examples are targeted mostly to the new users of the Toolkit, with the aim of helping them getting started with MUCM applications. Other proposed additions include Introductory pages, that introduce the Toolkit and can work as a road map to its methods, and Implementation pages, that

can hold all the implementation details of a Procedure that are involved enough not to fit in a Procedure page (e.g. treatment of numerical instabilities). A disclaimer page has also been prepared that should be placed next to each link to software or code that is uploaded in the Toolkit. Finally, a notation page will be prepared that will initially hold all the basic notation used so far, with the expectation of starting a discussion on the adoption of a uniform notation across the Toolkit.

Regarding the governance of the Toolkit, Tony O'Hagan has agreed to be the overall editor, and Yiannis Andrianakis will be the assistant editor. Each Thread will have an editor as well, and it is recommended that each Thread is edited by a PI. Each page will also have an author, who should be notified when other team members wish make changes to the their pages. The author of a page is also responsible for assigning a reviewer who will proofread and sign off their pages.

We finally propose a 3-month release cycle for the Toolkit, starting at the end of March 2009. This will allow 6 more releases until the end of MUCM, in October 2010. We also propose that the contributions in the releases constitute milestones, so as to ensure that the Toolkit is delivered on time and that the workload is spread evenly across the remainder of MUCM.