

Report on the development of the Toolkit

Y. Andrianakis and P.G. Challenor

October 13, 2009

1 Introduction

With the third release going live in October 2009, the toolkit has covered a fair amount of ground since its first release in March 2009. It has moved from 1 thread and 66 pages, to a whole 6 threads laid out in 116 pages. This document describes the current state of the toolkit, some of the major changes that have happened in its first three releases and some future directions.

2 Structure

The toolkit continues to be built around the model of threads. Threads provide a structured entry point to the toolkit, that aim to guide its user in accomplishing a particular MUCM application. The other page types are attached to one or more threads, with the exception of Meta pages that refer to the Toolkit as a whole. Analytically, the page types that are currently in use are shown in table 1.

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
Example	Exam	Pages showing an example of one or more concepts introduced in a thread
Meta	Meta	Any other page, not strictly part of the Toolkit's content, but rather referring to it.

Table 1: Page Types

2.1 Threads

2.1.1 Current Threads

The third release of the toolkit contains 6 threads. These are:

- **ThreadCoreGP** This thread treats the core problem, as set in DiscCore, using a Gaussian Process (GP) model of the simulator. In a narrative style, discusses and shows alternatives on setting up the GP emulator, selecting design points, fitting the model to the data, validating it and performing uncertainty and sensitivity analysis.
- **ThreadCoreBL** This is the sister thread of ThreadCoreGP, which also treats the core problem, but from a Bayes linear perspective.
- **ThreadVariantMultipleOutputs** This thread is a variant of the ThreadCoreGP, that extends the core problem by considering a simulator with multiple outputs.
- **ThreadVariantDynamic** This thread describes the construction of an emulator for a dynamic emulator, which is defined as a time evolving process.
- **ThreadVariantTwoLevelEmulation** The situation in which two models of the same physical process exist is described in this thread: a coarse and relatively inexpensive model and a more precise but also computationally more demanding. The thread considers the utilisation of information from the former for building a more accurate emulator for the latter.
- **ThreadGenericMultipleEmulators** This thread considers the situation in which two or more independent emulators have been built, each emulating either a single output of a simulator or outputs of different simulators altogether.

2.1.2 Future Threads

We envisage that the future releases of the toolkit will contain threads roughly on the following topics.

- A thread treating models with stochastic outputs
- A thread for models whose derivatives are observable
- A thread treating reality and model discrepancy

The above list may be modified/extended in due course.

2.1.3 Thread naming conventions

The existing threads follow a naming convention, according to which, their name consists of three parts:

$$Thread + [Identifier] + [Threadname]$$

Thread is the keyword denoting that the page is a thread. The identifiers used so far are *Core*, *Variant* or *Generic*. *Core* denotes a thread treating the core problem. *Variant* implies that the thread is a variant of another thread. The identifier *Generic* implies that the thread could be applied to any emulator, created with either a Core or a Variant thread. The naming convention tries to strike a balance between describing a thread precisely, while at the same time not being overly long.

A point raised in a discussion among team members, was that it would be useful to know which thread a variant thread is a variant of. This could be achieved by replacing the keyword variant with the name of the original thread. For example, ThreadVariantMultipleOutputs could be renamed

as ThreadCoreGPMultipleEmulators, or ThreadCoreDynamic could be renamed as ThreadCoreGPDynamic.

In a further effort to make the thread names more succinct, we could drop the identifier Generic, and then the generic character of a thread could be implied by the lack of an identifier. For example, ThreadGenericMultipleEmulators could be changed to ThreadMultipleEmulators.

2.2 Examples

In the second release a new page type was introduced, the Example page. The toolkit currently contains two examples, ExamCoreGP1Dim and ExamCoreGP2Dim. The examples follow the development of ThreadCoreGP, showing how an emulator can be fitted to a number of simulator runs, how it can be validated and how to perform an uncertainty and sensitivity analysis.

Example pages were proposed in the midterm meeting as a tool for a hands on introduction to the methods considered in the toolkit. While at the moment only examples for the ThreadCoreGP exist, there are plans for writing examples for other threads, starting with ThreadCoreBL in release 4.

2.3 Deprecated page types

Although included in previous plans for the development of the toolkit, the following pages types are no longer used

- **Templates** Suggested templates for other pages
- **Implementation** Pages designated for holding implementation details. Implementation details are discussed in Discussion pages, as in DiscBuildCoreGP
- **Subject** Subject areas are no longer used, as they have been superseded by Threads.

3 Notation

Prior to the third release, the MUCM team has put together a considerable effort in standardising the notation used in the toolkit. Noting that an exhaustive documentation of the notation would be both impractical and of limited value, it was decided that the notation will be fixed for a small number of fundamental quantities that are used throughout the toolkit. This formed the *reserved* notation, which is documented in the MetaNotation page, also available in the public domain. Apart from the reserved notation, MetaNotation lists the notational conventions are used throughout the toolkit, such as conventions for matrices and vectors, functions, expectations etc.

Having a small set of reserved symbols offers a greater flexibility in choosing symbols that are used in a small number of pages in the toolkit. These locally defined symbols form the *unreserved* notation. This either has to be defined locally in a page, or a pointer to another page with its definition must be given. The unreserved notation is not globally documented, and cannot overlap with the reserved notation.

4 Governance

The experience of the three first releases has shown that one PI is responsible for the overall contents of a thread. The pages that consist the thread are then written either by the PI himself or by a RAPS member with the necessary expertise.

Tony O'Hagan signs off all the pages and provides comments and feedback. This is on one hand desirable, because it ensures that the toolkit has a uniform character throughout. On the other hand, it requires a significant commitment in terms of time on behalf of the toolkit editor. We think that having a second person signing off the pages, apart from the editor, could decrease his workload, especially in terms of checking the accuracy of the algebra and procedures described in the toolkit.

5 Toolkit Technologies

It is generally accepted that the quality with which the mathematical content of the toolkit is rendered is suboptimal. The most likely reason for this is that html cannot recognise the size and baseline of the equation fonts, which are saved as images, and therefore their scaling and placement is not always in harmony with the rest of the text. Methods for improving this condition will be looked at in the near future.

We are very much interested in the opinion of our users and this is why we are asking for their feedback. At the moment, users are encouraged to send their feedback by mailing the team. We are planning to replace this with the use of forms where users can submit their views or comments. Additionally, we will also look into a web analytics system that will help measure the usage of the toolkit.

6 MetaToDo page

With less than a year to go before the end of MUCM, and 4 remaining releases of the toolkit, it is important that the tasks that need doing before the last scheduled release of the toolkit in October 2010 are documented. For this reason the MetaToDo page was created, where the Toolkit authors are encouraged to record the tasks they have agreed to undertake, so as to help their timely completion.

7 Release cycle

So far there have been three releases of the toolkit (Mar 09, Jul 09, Oct 09). We are planning to continue with a three month cycle, which will allow 4 more releases until the end of the project.