Explaining Models in Software and Text

Murray, N.I.[1,2] and **S.M. Cuddy[1,2]**

[1]CSIRO Land and Water, [2]CRC for Catchment Hydrology, E-Mail: nicholas.murray@csiro.au

Keywords: *modelling**,** documentation, guidelines, explanation**.***

EXTENDED ABSTRACT

The building of models is a common way of capturing and representing knowledge about some aspect of the 'real' world. In most cases it is not possible (nor perhaps necessary) to capture all the complexities and interactions inherent in the system. These are reduced to a manageable (and still meaningful) abstraction through a series of assumptions and simplifications that reflect biases, experience and expertise of the model builder. If the models are for the use of others, such information has commonly been transferred via exposure in domain-specific and sometimes documented in technical reference manuals.

With the move to build models that are intended for other purposes (eg components in a larger, compound model intended for a range of audiences) comes the need to not just document, but explain, how to use the model, how to connect it to other models, and how to interpret the results. Such explanation has many vehicles – more traditional forms such as manuals and guides - and through appropriately designed user interfaces and on-line help.

The paper differentiates between model developer, model deliverer, and model user to assist with categorising the information flows (and explanatory needs) between them. We propose a simple classification of explanation, into "up-front" and "community-generated". The former is more traditional and refers to documentation that anticipates the user needs and is distributed with the software. The latter is where the developer or deliverer forms a user group that 'grows' its own knowledge about the nuances of the model and its application.

In this paper we critique the explanatory power of the documentation provided with the CRC for Catchment Hydrology's Environmental Management Support System (EMSS) (Vertessy *et al*., 2001). Rather than examine the documentation per se (ie the up-front), we have analysed the queries that have come back to the developers through the user group email facility (community-generated). While this approach is influenced by the communicative nature of the users, it does offer a rare opportunity to gauge user sentiment without the use of questionnaires.

The investigation considered both quality and timeliness of response, and the usefulness of the community-generated explanation. It was sufficient for us to assert that poor documentation is harmful to developer/deliverer/user relationships. More attention must be given to the content of up-front documentation. As documentation is often considered an onerous task by model developers, we propose a minimum set of guidelines for documenting models, both in text and in software, that we consider necessary and not overly demanding.

.

1. INTRODUCTION

Models encapsulate and transfer knowledge about specific processes and relationships.

By their nature, models incorporate assumptions and simplifications of processes due to time, computation or complexity constraints. Model users need to be familiar with these assumptions and constraints in order to use models effectively (to produce accurate results) and appropriately, using a model that is meaningful in the problem domain.

Previously, "appropriate use" fell outside of the responsibility of the model developer (ie the researcher), but as models are used increasingly as components of larger, compound models, the choice of a model may depend on which other models have already been chosen. This requires the model developer to explain, either via documentation, or programmatically - via a model's built-in checking mechanism - how to use the model, how to connect it to other models, and how to interpret its results.

In this paper, we examine a subset of the areas where there has been either inadequate explanation, or a failure of explanation, in using hydrologic models from the CRC for Catchment Hydrology's product Environmental Management Support System (EMSS), which is a framework used for constructing compound models. We attempt to highlight areas where there is a clear need for better explanation, and we then describe the derivation of a set of guidelines for documenting models both in text and in software.

2. WHAT IS EXPLANATION?

Explanation is defined by the Oxford English Dictionary as "That which explains, makes clear, or accounts for..." We define explanation to include "assistive devices" that convey *meaning* to model users. This includes traditional forms of information, such as user manuals, journal papers, reference guides, and online resources. It also incorporates the design of user interfaces to both accept input and convey results, and to educate the user not only about modelling semantics, but how and why a system produces certain results. Moulin *et al.* (2002) points out that users are far more likely to trust a system when they can see how the system has produced a particular answer. Well-written error messages generated within a model, for example, are a simple way in which the model can inform a user that a particular design choice is not correct.

3. THE ROLE OF EXPLANATION

The role of explanation in modelling is to clarify and assist the user in the appropriate and meaningful use of a model, *in the absence of the model developer*. This last point is important, as the community of model users is nearly always larger than the community of model developers and it would be too time-consuming for model developers to attend to every request for help.

In order to discuss the issues surrounding model use, we must first clarify the different roles that people and/or organisation play in the modelling process. We will use the following definitions:

3.1. Model Developer

A model *developer*, usually a researcher, develops a model, either as part of a fundamental research effort or in response to a request by another party (policy developers, stakeholders, government agencies, and so on). Model developers have extensive background knowledge of the problem domain. They will also be familiar with modelling assumptions & tradeoffs and uncertainty issues surrounding the model.

3.2. Model Deliverer

A model *deliverer* takes the model from the model developer, perhaps as a journal paper, or algorithm, and wraps the model in software so that it can be used more widely.

Model deliverers are usually software developers, who package the model algorithm into a form that exposes aspects of the model - model inputs, outputs and specified internal variables - to a model user. The delivery package could be client software (installed locally on a computer), a web-based system (for remote processing, or possibly using data from remote sources) or a database. The delivery package could also include processing, analysis and visualisation capabilities that serve to clarify, explain, extend or make relevant the model outputs.

3.3. Model User

A model user is a person or organisation who takes a model and uses it for a specific application. Model users comprise those people who work with models or sets of connected models, running scenarios, perhaps applying different sets of input data, and interpreting the results.

Model users are often third parties, performing work on behalf of their clients, and include consultants, federal and state agencies.

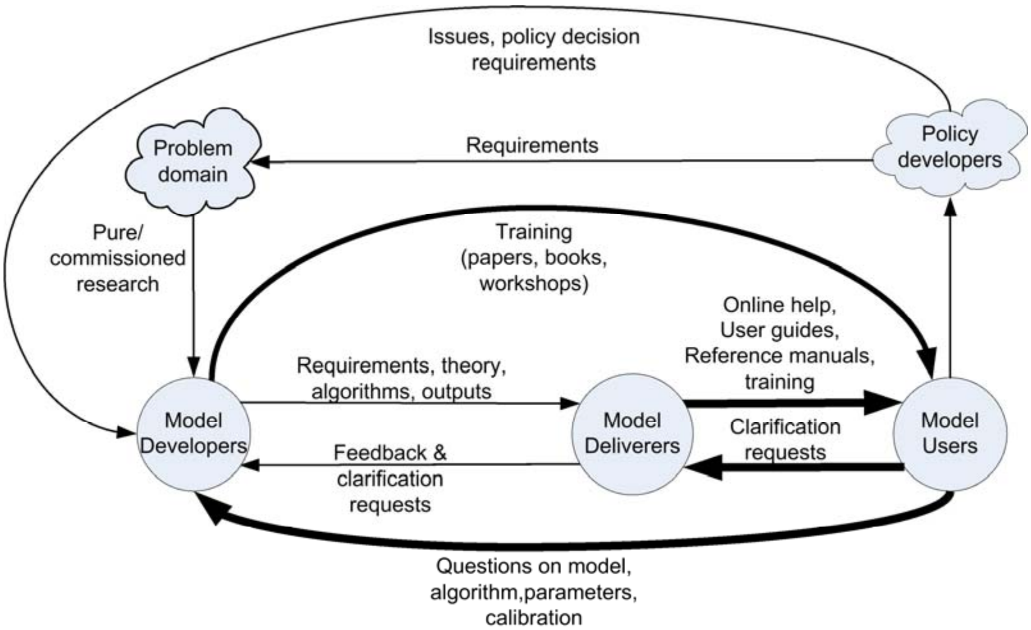

Figure 1: Documentation flow for model developers, deliverers, and users. The information flow paths considered in this paper, are in **bold**.

The different roles and information flows between the users are illustrated in Figure 1. Note that this is in itself an idealised model, and that we are primarily concerned with model users' interactions with model deliverers and model developers.

4. TWO POSSIBLE APPROACHES TO DOCUMENTATION

4.1. Community-generated explanation

The community approach assumes that there is some form of initial model documentation, perhaps given in a paper or user manual. The model developer or deliverer encourages users to form a group and "grow" their own knowledge as members use the model. The advantages to this approach are that the group can quickly gain knowledge in areas that are most relevant and familar, and the model developer is not called upon to support each request for help. The disadvantages are that determining the answer to a question could take several days or weeks depending on the availability of people with the required knowledge (including the model developer), and answers could also be inaccurate or inappropriate.

4.2. Up-front explanation

In this approach, the model developer thoroughly documents all aspects of the model, from its theoretical basis to its implementation in a software product.

The advantages of this approach are that documentation can be built-in as part of a project's funding base and the model developer and deliverer can concentrate on anticipating and answering questions about the model. However, there are disadvantages in generating "up-front" explanation without first considering how users will operate and use the model (or reflecting on prior experience). The authors may end up documenting subjects that are not representative of users' true needs. Creating explanatory systems (including documentation) that incorporates any kind of model of user needs and behaviour is time-consuming and difficult (Delisle and Moulin, 2002). It has been shown that task modelling, and using task modelling tools such as TAMOT (Lu *et al.*, 2002), can assist with both helping the user develop an understanding of a system, and by giving the system developers a better idea of how users interact with, and use, software modelling systems, however no task modelling was done when EMSS was being developed.

5. ANALYSING THE EMSS EMAIL LIST

5.1. Methodology

In order to discover the types of information that users were requesting, we analysed 248 postings to the EMSS "Development Projects" email list. We divided the postings into five major categories, described in Table 1.

Table 1: Email posting categories. Percentage values are of relevant postings ie after "Miscellaneous" postings were removed.

Category	Category contents	Posts
Data-related	Data requirements, preparation, units, coverage	30 (17%)
Documentation-related	Clarification, missing steps, procedures to do certain tasks, references,	38 (21.6%)
Model-related	Conceptual issues, parameterisation, calibration	33 (18.8%)
Software-related	Bug reports, capabilities and/or limits, feature requests, output & file storage issues, usage and/or usability issues, announcements of new versions	75 (42.6%)
Miscellaneous	Non model-related discussion, thanks, workshop announcements	70

After removing emails in the Miscellaneous category, and announcements of new software and documentation, we divided the remaining total of 176 emails into "requests" for help and "answers" to those requests. It was useful to classify the requests and answers into subcategories, listed in Table 2.

Table 2: Email posting subcategories. Posts show relevant emails ie after "Miscellaneous" postings were removed.

Subcategory	Subcategory contents	Posts
Clarification requested	User seeks an explanation of a model or a document. User may consider existing material on the topic incomplete or insufficient.Also includes proposals or ideas put forward for comment by others.	72 (40%)
Clarification provided	New explanatory material, or an explanation, expansion, clarification or addendum to material already provided	67 (38%)
Clarification request for prior request	User seeks clarification on a response previously posted to the list.	8 (5%)
Clarification provided for prior response	Further clarification on a previous response, either adding new information or qualifying or validating information that was previously given.	15 (9%)

Finally, we also classifed emails as being either from people or organisations representing the "model using community" (CTY) or from people or organisations identified as "model developers or deliverers" (MDD).

5.2. Data Limitations

There were several limitations in the both the survey methodology and data.

We did not consider whether requests were answered to the satisfaction of the requester (though anectodotally most requests were fulfilled). Our interest lay in the types of requests being made, and the subjects requiring further explanation.

If there was more than one request or answer in a single email, we considered only the first request or response, as it was considered that the first request/response would have been the most important. We have not yet analysed delays between requests and answers.

The population surveyed was limited, and comprised people and organisations who were already CRC for Catchment Hydrology associates, and thus quite familiar with the problem domain. so the "knowledge gap" between question and response was not large.

Finally, the survey did not consider requests, responses and clarifications made by other means, including phone conversations, meetings or during the three EMSS training workshops held during the survey period of 28 March 2003 - 10 June 2005.

6. RESULTS

6.1. General

Tables 1 and 2 show the number of relevant posts to the list (after non-model-related posts were removed). Figure 2 shows that, the ratio of requests (Clarification requested) to answers (Clarification provided) was almost 1:1. While not shown here, the actual answers posted to the list were well-matched with requests, indicating that most requestors received a response of some kind. Relatively few people (5%) needed additional explanation of a prior response. CTY users made 73% of the posts to the list, leading us to hypothesise that CTY members were able to help each other for about half of the requests - quite encouraging. However, further analysis showed that of the 34 individuals who posted to the list, only 6 individuals (4 CTY, 2 MDD) posted 10 or more items (whether relevant to modelling or not). Most of the requests and community support came from these 4 CTY users. On the MDD side, two individuals provided most of the responses to the list. This points to a relatively heavy, and unsustainable, software support workload borne by a small number of people.

Figure 3 shows the breakdown of requests and answers, as a percentage of the 176 emails, by category.

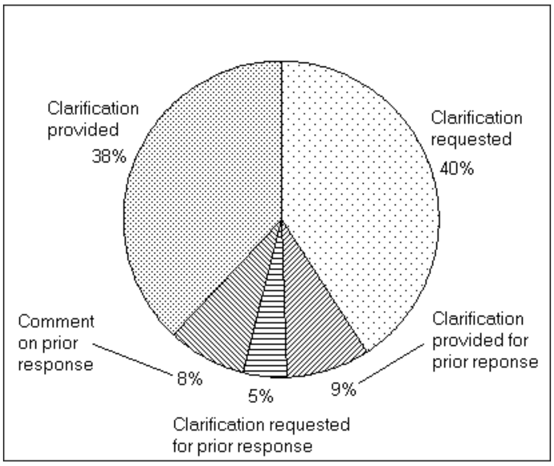

Figure 2: Requests and clarifications

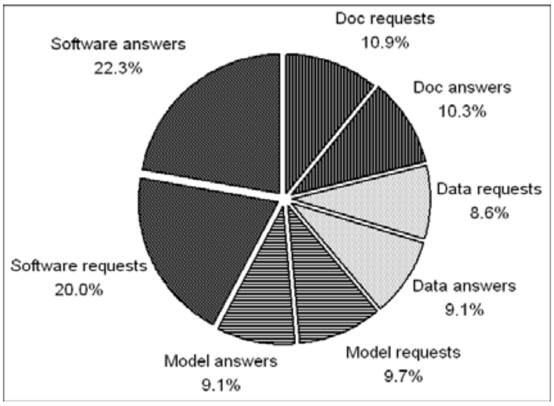

Figure 3: Requests and answers by category

We could not determine the reasons for the relatively low level of interaction of the other 28 email list participants. It is possible that they did not encounter problems in the same areas as other participants, that previous requests and responses met their needs, or that their support requirements were met elsewhere. The "community engagement" aspect requires further research.

6.2. Data Requirements

8.6% of total requests dealt with data collection and preparation. This included data units, file formats, finding sources of data and converting data into formats compatible with EMSS. It also included requests concerning pre-processing steps for data, most commonly for DEMs and rainfall data, and in some cases, the physical locations on disk required for data files. In both the EMSS email list, and informally, users stated that data gathering and format issues were a major concern.

6.3. Documentation

10.9% of requests were for more documentation of various usage aspects of EMSS, primarily:

- data processing steps that were not documented or incompletely documented
- using external data processing tools
- clarifications of statements or procedures in the EMSS User Guide

This category includes requests where the emphasis was on documentation, though the request itself could have been related to model or software usage issues.

EMSS had both a user manual and an HTML-based resource called the "EMSS Assistant" (Figucio *et al*, 2003). The EMSS Assistant was intended to provide information on the EMSS models, background information on the catchments in which the EMSS had been applied, reference material concerning the built-in models, and data requirements and formats. Unfortunately, much of the material was outdated or non-existent, poorly maintained, rarely-used or not relevant to users, incomplete or required the reader to jump from a web browser to PDF and back again: Nielsen (2003) describes in detail the usability problems when mixing web-based and PDF-formatted material.

6.4. Model-related questions

There were three subcategories for model-related issues:

- clarification of algorithms, including explaining parameters and their valid ranges, which parameters to use or modify, weighting factors and applicability of model algorithms to a particular problem
- calibration of models
- interaction between models

The greatest number of model-related requests, 9.1% of the total, was for clarification of algorithms. In some cases, original published papers, books or workshop notes did not contain sufficient information, contained errors, or the process for using the information was not clear.

6.5. Software

There were five subcategories of software request:

- bug reports or requests for an update on a previously reported bug
- capability clarifications, where a user inquires if EMSS has a particular feature

- feature requests, that is, a request for additional capability
- software output clarifications, where users have questions about what the software is producing, showing or visualising
- software usage issues, where users want to know how to perform various processing steps, find options, or have questions about the user interface or workflow

The most common request concerned software usage issues (12.6%). The community support approach worked "well", insofar as there were an almost equal number of responses (11.4%). The other subcategories totalled only 7.4%, indicating that usage issues ("How do I ...") were the biggest CTY problems with EMSS.

Requests for new features were a minor part of software-related queries due to an EMSS workshop in December 2003 where features and software bugs were the main agenda items.

However, the email posts point to many existing features in EMSS being inadequately documented, confusing user interface elements, and behaviour that deviated from behaviour that most users would expect from a Windows application.

7. DISCUSSION

While the "community-generated" approach generated a great deal of information and interaction on the mailing list, it was obvious that there was room for improvement in each of the four categories:

- Software-related (usability and workflow)
- Data-related (requirements and pre-processing)
- Model-related (algorithms and theory)
- Documentation-related (processes and coverage)

During the development of the successor to EMSS, the application E2, the E2 development team acquired anecdotal evidence regarding EMSS' usability and data requirements. E2 was constructed with a different modelling framework: TIME (Rahman *et al*, 2003), and is described in Perraud *et al* (2005). E2 has a more Windows-compliant interface, which addresses many of the usability issues. It also supports a greater variety of data input and export formats using the underlying TIME framework.

The EMSS User Guide had been revised several times during the life of EMSS, but we believe that it failed in adequately explaining or conveying all the information necessary for EMSS users. The problems in documentation were both structural, through having two different sources of information spread over several different file types (text, HTML, Flash, PDF and Microsoft Word), and pragmatic: the documentation did not satisfy EMSS users' operational and processing needs. This was obvious from the result that two of the MDD contributors provided the bulk of the clarifications to the email list.

There was a clear need to review and consolidate the existing material, to document processes and steps that users requested time and again (such as calibrating models) and also to ensure that model developers captured essential information when describing their models, as this would reduce the workload on the MDD participants.

This led to much of the EMSS Assistant content being folded into the EMSS User Guide. However, the material was still not in a form that was accessible to many users, and concentrated too much on basic operations, such as the operation of EMSS' underlying modelling framework Tarsier (see Rahman *et al*., 2004) than on usage reflecting EMSS users' needs. Moving EMSS scenarios and data files from one location on one computer to another, for example, was a poorly-documented process until one email-list user developed and posted his own series of steps for others to use.

We believe that many of these issues with EMSS stemmed from insufficient up-front information on how EMSS would be applied, and the tasks that EMSS users would perform.

8. IMPLEMENTATION - MODELLING GUIDELINES

To address the model algorithm, documentation and data preparation issues, the E2 development team identified the deficiencies in the existing documentation and created a series of guidelines "up-front" for documenting models (Murray *et al*, 2005). These guidelines listed and described, in a generic way, many of the possible aspects of hydrologic models, including input data sources, formats and requirements, model processes and algorithms, testing, calibration and spatiotemporal validity constraints, the theoretical basis for the model, and uncertainty and sensitivity characteristics. The intent was to provide model developers with a template where they could fill in the types of information most commonly asked requested. The initial released version of this "description" document was the E2 Component Model Reference (Argent *et al* 2005a). This document was a first step towards satisfying the requirements, usage patterns and approaches of model users.

Also, a document describing the different file formats supported by E2 was developed (Argent *et al* 2005c).

9. FUTURE WORK

Our priority is to develop ways of measuring the effectiveness of methods of explanation. This includes monitoring the E2 mailing list to detect any improvements or trends in the types of requests that users make, using the data from the EMSS mailing list as a baseline. It also includes task modelling to discover how people interact with, and use, the application.

There is also scope for "instrumenting" the E2 application (by recording user actions, detecting errors, and tracking requests to the help system) to gather more data on how the application is used.

We will also examine ways in which we can increase the level of community engagement with products such as EMSS or E2. In addition to reducing the support load on the model developers, many of the improvements made to the documentation and software were a result of community requests and suggestions.

10. CONCLUSIONS

In this paper, we have examined some of the problems surrounding explanation of modelling concepts as embedded in software applications, with specific examples from EMSS and the EMSS mailing list. Though many of the issues were suspected *a priori*, we felt it important to discover the relative proportion and nature of the explanation problems so that we could attempt to remedy them in future products. The development and release of EMSS' successor, E2, has partially addressed the issues discussed above, but more work is required to evaluate the effectiveness of the changes from a user's perspective.

11. ACKNOWLEDGMENTS

The authors acknowledge the assistance of the EMSS and E2 development teams during the preparation of this paper.

12. REFERENCES

Argent, R.M., G.D. Podger, J-M. Perraud and N. Murray, (2005a), *E2 Component Models Reference*, CRC Catchment Hydrology, 64pp

Argent, R.M., G.D. Podger, R. Grayson, and K. Fowler. (2005b), *E2 User Guide*, CRC for Catchment Hydrology, 100pp

Argent, R.M., G.D. Podger, (2005c), *Toolkit Data File Formats*, CRC Catchment Hydrology, 19pp

Delisle, S., and B. Moulin, (2002), User Interfaces and Help Systems: From Helplessness to Intelligent Assistance, *Artificial Intelligence Review* 18, 117–157

Figucio, D., S.M. Cuddy and F. Marston, (2003), *EMSS Assistant* (component of the EMSS).

Grayson, R. (2005), Questions on E2 from the Catchment Modelling School 2005 workshops, Personal communication, 9 August 2005.

Lu, S., C. Paris and K. Vander Linden, (2002), Tamot: Towards a Flexible Task Modelling Tool . In *The Proceedings of Human Factors 2002*, Melbourne, Australia, November 25-27.

Moulin, B., H. Irandoust, M. Bélanger and G. Desbordes, (2002), Explanation and Argumentation Capabilities: Towards the Creation of More Persuasive Agents, *Artificial Intelligence Review* 17, 169–222.

Murray, N., G.D. Podger, S.M. Cuddy and A. van Dijk, (2005), *Describing Models*, CSIRO Land and Water internal working document (unpublished).

Nielsen, J., (2003), *PDF: Unfit for human consumption*, http://www.useit.com/alertbox/20030714.html [Accessed 28 August 2005]

Perraud, J-M., S.P. Seaton, J.M. Rahman, G.P. Davis, R.M. Argent and G.D. Podger, (2005), The Architecture of the E2 Catchment Modelling Framework, In: *Proceedings of MODSIM 2005*, December 12-15, Melbourne, Australia. MSSANZ, Canberra.

Rahman, J.M., S.M. Cuddy and F.G.R. Watson, (2004), Tarsier and ICMS: two approaches to framework development. *Mathematics and Computers in Simulation* 64, 339-350.

Rahman, J.M., S.P. Seaton, J-M Perraud, H. Hotham, D. Verrelli and J.R. Coleman, (2003), It's TIME for a New Environmental Modelling Framework, In: *Proceedings of MODSIM 2003*, July 14-17, Townsville, Australia. MSSANZ, Canberra.

Vertessy, R.A., F.G.R. Watson, J.M. Rahman, S.M. Cuddy, S.P. Seaton, F.H. Chiew, P.J. Scanlon, F.M. Marston, L. Lymburner, S. Jeannelle and M. Verbunt. (2001), New software to aid water quality management in the catchments and waterways of the south-east Queensland region. In: *Proc. 3rd Australian Stream Management Conferenc*e. Brisbane, pp 611-616. CRCCH, Melbourne.