I/O Associations in Scientific Software: A Study of SWMM

Zedong Peng[†], Xuanyi Lin[†], Nan Niu[†], and Omar I. Abdul-Aziz[‡]

[†]University of Cincinnati, Cincinnati, OH, USA 45221
[‡]West Virginia University, Morgantown, WV, USA 26506
{pengzd,linx7}@mail.uc.edu, nan.niu@uc.edu, oiabdulaziz@mail.wvu.edu

Abstract. Understanding which input and output variables are related to each other is important for metamorphic testing, a simple and effective approach for testing scientific software. We report in this paper a quantitative analysis of input/output (I/O) associations based on co-occurrence statistics of the user manual, as well as association rule mining of a user forum, of the Storm Water Management Model (SWMM). The results show a positive correlation of the identified I/O pairs, and further reveal the complementary aspects of the user manual and user forum in supporting scientific software engineering tasks.

Keywords: Scientific software, user manual, user forum, association rule mining, Storm Water Management Model (SWMM).

1 Introduction

The behavior of scientific software, e.g., a seismic wave propagation [11], is typically a function of a large input space with hundreds of variables. Similarly, the output space is often large with many variables to be computed. Rather than requiring stimuli from the users in an interactive mode, scientific software executes once the input values are entered as a batch [32].

The large input/output (I/O) spaces are common for the scientific understanding of complex phenomena like climate change. However, the size and complexity have been recognized as challenges for software testing [15], especially for selecting test cases from a large input space and for determining the corresponding outputs to examine.

Relating I/O is fundamental to metamorphic testing, which is considered to be a simple and effective approach for testing scientific software [8]. The prototypical example is the trigonometric function: sine(x) [13]. The exact value of sine(x) may be unavailable due to floating-point computations. Metamorphic testing uses properties like sine(x)=sine($\pi-x$) to test any implementation without having to know the concrete values of either sine(x) or sine($\pi-x$).

While the I/O relations are clear in the above example, namely, changing the input of an angle relates to the output of the angle's sine value, determining the I/O associations at the system level, rather than at the unit level, is difficult due to the size, complexity, and batch execution mode. The scientific software of our

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_29

study, for example, has over 800 input variables and over 150 output variables. Tracking the I/O dependencies in the source code (e.g., via program slicing or define-use data relationships) can face scalability issues.

In this paper, we investigate I/O associations in the user manual and user forum of a scientific software system: the Storm Water Management Model (SWMM) [30] developed and maintained by the U.S. Environmental Protection Agency (EPA) for five decades. We manually identify the I/O variables from the SWMM user manual [26], and analyze their degrees of association based on the co-occurrence statistics. We further mine the I/O variables' association rules from one of the largest SWMM user forums with approximately 2,000 contributors and 17,000 posts [21]. Comparing the I/O associations reveals the complementary aspects of the user manual and the user forum, suggesting concrete ways to exploit metamorphic testing for scientific software's quality assurance.

The contributions of our work lie in the quantification of I/O associations from how the scientific software is introduced by the development team to the users, and how the end users discuss the actual software usages among themselves. In what follows, we provide background information and introduce SWMM in Section 2. Section 3 presents our quantification and comparison of the SWMM I/O associations, Section 4 discusses the implications of our results, and finally, Section 5 draws some concluding remarks and outlines future work.

2 Background

2.1 Metamorphic Relations and I/O Associations

Metamorphic testing requires properties like sine(x)=sine($\pi-x$) to guide the testing process. These properties represent necessary conditions for the software to behave correctly, and are referred to as *metamorphic relations* (MRs). Each MR consists of two parts: (1) an input transformation that can be used to generate new test cases from existing test data, and (2) an output relation that compares the outputs produced by a pair of test cases. As shown in Figure 1, establishing an MR is about connecting a particular input with a corresponding output, and then asserting how such an I/O pair co-changes.

Constructing MRs is an essential task in metamorphic testing. The early work by Chen *et al.* [4], for example, relied on researchers' domain knowledge to manually create one MR and further illustrated the MR's effectiveness via testing a program that solves an elliptic partial differential equation with Dirichlet boundary conditions. Murphy *et al.* [17] made one of the first attempts to enumerate six MR classes applicable to numerical and collection-like inputs.

Although numerical MRs may be suitable for computational units like the trigonometric functions, system testing in which the scientific software is tested as a whole likely requires different MRs. Our work on integrating two different scientific software systems [7, 12, 14], for instance, shows the importance of understanding the entire software's inputs, outputs, and their relationships. Next is an introduction of the scientific software whose I/O associations are the focal points of our study.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_29

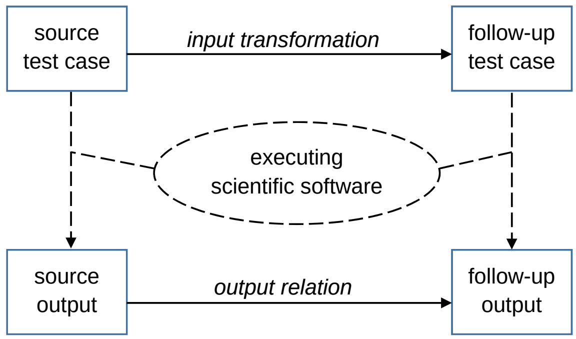


Fig. 1. A metamorphic relation (MR) consists of an *input* transformation (e.g., from x to $\pi - x$) and the associated *output* relation (e.g., equivalence relation).

2.2 Storm Water Management Model (SWMM)

The Storm Water Management Model (SWMM) [30], created by the U.S. Environmental Protection Agency (EPA), is a dynamic rainfall-runoff simulation model that computes runoff quantity and quality from primarily urban areas. The development of SWMM began in 1971 and since then the software has undergone several major upgrades.

The most current implementation of the model is version 5.1.015 which was released in July 2020. Figure 2 shows a screenshot of SWMM running as a Windows application. The computational engine, which implements hydraulic modeling, pollutant load estimation, etc. is written in C/C++ with about 46,300 lines of code. This size is considered to be medium (between 1,000 and 100,000 lines of code) according to Sanders and Kelly's study of scientific software [27].

The users of SWMM include hydrologists, engineers, and water resources management specialists who are interested in the planning, analysis, and design related to storm water runoff, combined and sanitary sewers, and other drainage systems in urban areas. Thousands of studies worldwide have been carried out by using SWMM, such as land use [1, 6] and stormwater modeling [3].

3 I/O Associations in SWMM

The wide adoption of SWMM in supporting critical tasks of urban planning and environment protection makes it important for the development team at EPA to introduce the software to its users via a user manual [26]. In fact, producing the user manual is not only a common practice among scientific software developers [18], but also a requirement mandated by agencies like EPA [29] and the U.S. Geological Survey (USGS) [31]. For software evolved over many years, the documentation generated by end users themselves, such as user forums, builds a massive resource which has gradually become informative and comprehensive [22]. This section thus reports our analysis of SWMM's user manual in Section 3.1

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_29

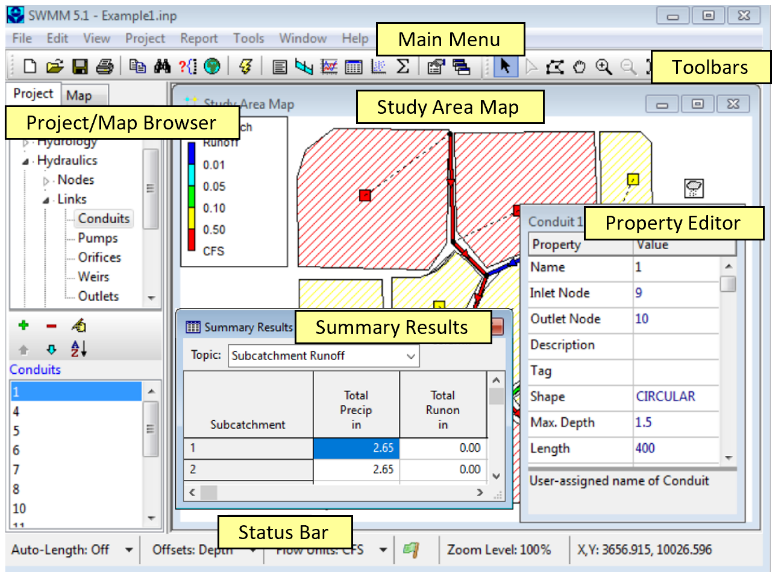


Fig. 2. SWMM running as a Windows application, annotated with functional areas in the graphical user interface.

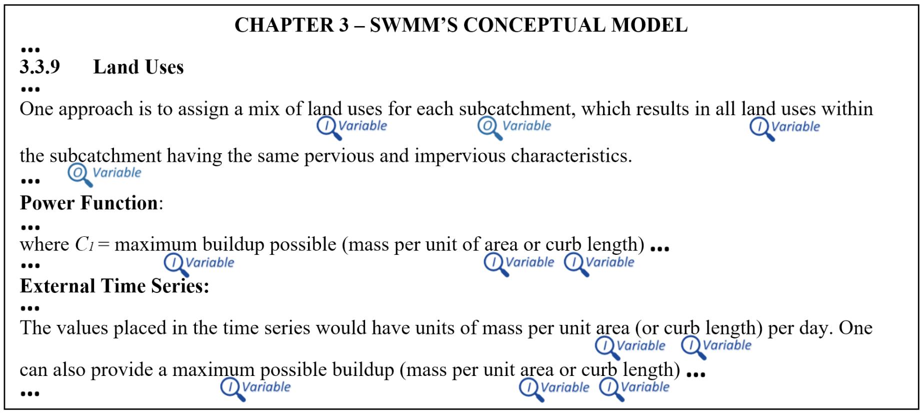
and a user forum in Section 3.2. We then compare the I/O associations from these sources in Section 3.3, and discuss the threats to validity in Section 3.4.

3.1 Co-Occurrence Statistics in User Manual

The SWMM user manual (version 5.1) is a 353-page PDF document written by a core developer and environmental scientist at EPA [26]. It contains 12 chapters and 5 appendices, covering software installation and configuration steps, SWMM's conceptual model, working with map and objects (e.g., conduits of Figure 2), running a simulation, viewing results (e.g., subcatchment runoff summary of Figure 2), and detailed information about units of measurement, properties of visual objects, and error and warning messages. The user manual is such a comprehensive document that it remains relevant for the different sub-versions of SWMM 5.1 (5.1.010–5.1.015) since 2015.

Building on the recent work [24], we manually identified the I/O variables from SWMM's user manual. Two researchers independently performed the variable identification in a randomly chosen chapter, and Cohen's kappa between their results was 0.87 indicating an almost perfect agreement [5]. We attribute this high inter-rater agreement to the clarity of SWMM's user manual. The two researchers then individually identified the variables for the rest of the user manual. In total, 807 input and 164 output variables were identified and the manual work took approximately 40 human-hours; however, this one-time cost would be amortized over subsequent co-occurrence analysis and association rule mining.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_29



CHAPTER 3 – SWMM'S CONCEPTUAL MODEL

...

3.3.9 Land Uses

...

One approach is to assign a mix of land uses for each subcatchment, which results in all land uses within the subcatchment having the same pervious and impervious characteristics.

...

Power Function:

...

where C_1 = maximum buildup possible (mass per unit of area or curb length) ...

...

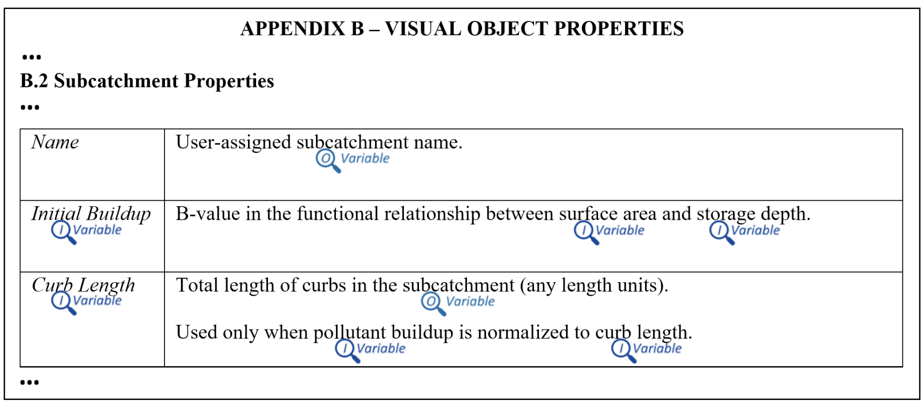
External Time Series:

...

The values placed in the time series would have units of mass per unit area (or curb length) per day. One can also provide a maximum possible buildup (mass per unit area or curb length) ...

...

(a) I/O variables in natural language text



APPENDIX B – VISUAL OBJECT PROPERTIES

...

B.2 Subcatchment Properties

...

Name	User-assigned subcatchment name.
Initial Buildup	B-value in the functional relationship between surface area and storage depth.
Curb Length	Total length of curbs in the subcatchment (any length units). Used only when pollutant buildup is normalized to curb length.

...

(b) I/O variables in table

Fig. 3. Excerpts of SWMM's user manual [26], annotated with I/O variables.

We also share the data of our work, including the I/O variables, in the institutional digital preservation site Scholar@UC [19, 23] to facilitate replication.

Figure 3 shows the excerpts of SWMM's user manual, annotated with the input ('I') and output ('O') variables. To explore the I/O associations, we distinguish their appearances in the natural language texts (cf. Figure 3a) and in the structured tables (cf. Figure 3b). We measure the extent to which an input variable is discoverable together with an output variable as follows.

- Natural language text is hierarchical: a chapter has one or more sections or sub-sections, a section or sub-section has one or more paragraphs, and a paragraph has one or more sentences. We therefore use the hierarchical information to calculate how closely related a pair of I/O variables are to each other. On one hand, if *all* the co-occurrences are within a sentence, then we

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_29

consider the I/O pair to be strongly associated. On the other hand, if no co-occurrences are observed within the same sentence, paragraph, section/sub-section, or chapter, the I/O variables are loosely associated. To illustrate the degree of association calculation, let us consider the input variable "curb length" and the output variable "subcatchment" in Figure 3a. The number of co-occurrences of this pair is 2 in the sub-section of §3.3.9. This is because we take the minimum count between "curb length" (3 times) and "subcatchment" (2 times) in Figure 3a. In the entire user manual, the number of co-occurrences of "curb length" and "subcatchment" in a sentence, paragraph, section/sub-section, and chapter is 3, 3, 16, and 16 respectively. We compute the ratios of sentence over paragraph ($\frac{3}{3}$), paragraph over section/sub-section ($\frac{3}{16}$), section/sub-section over chapter ($\frac{16}{16}$), and then take the average of the three ratios (0.729) as this I/O pair's association degree in the natural language part of the user manual.

- Tables like Figure 3b provide structured ways to relate an input variable and an output variable. We therefore count the number of tables in which an I/O pair co-appears, and then divide it by the total number of tables the user manual has as an implication of how the pair of I/O variables may be structurally associated together. This calculation leads to a $\frac{1}{107}$=0.009 degree of association between "curb length" and "subcatchment" in the tabular part of the user manual.
- We combine the natural language part and the tabular part by taking the average of the above two measures. Thus, the association of "curb length" and "subcatchment" in the user manual is $\frac{0.729+0.009}{2}$=0.369.

Our rationale is to estimate how easy a user would find a pair of I/O variables being related in the user manual. By employing WordNet's lemmatizer (wordnet.princeton.edu) to convert words into the inflected roots (e.g., "conduits" to "conduit"), we rank SWMM's I/O pairs based on the degrees of association. Table 1 lists the ten top-ranked pairs and shows their associations in the natural language part, the tabular part, and the user manual as a whole. More complete results can be found in our online data [23].

Table 1. I/O associations based on variable co-occurrences in SWMM's user manual.

rank	**input variable**	**output variable**	**textual part**	**tabular part**	**user manual**
1	rain barrel	runoff	1.000	0.000	0.500
2	conduit	hours flooded	1.000	0.000	0.500
3	conduit	peak depth	1.000	0.000	0.500
4	conduit	peak runoff	1.000	0.000	0.500
5	aquifer	runoff	1.000	0.000	0.500
6	rainfall	hours flooded	1.000	0.000	0.500
7	outlet	flow routing	0.952	0.000	0.476
8	wet step	runoff	0.889	0.000	0.444
9	node invert elevation	depth	0.889	0.000	0.444
10	dynamic wave	flow	0.861	0.009	0.435

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_29

3.2 Association Rule Mining in User Forum

While SWMM's user manual is written by one scientific software developer, a forum like Open SWMM [21] records the questions, discussions, and interactions of thousands of SWMM users. The typical topics include how to install, configure, and run the software. The experience of running the software leads some users to post their frustrations about executions producing no result, their doubts about the validity of the generated results, and their dissatisfactions about the execution process. Sometimes, others respond to these questions to clarify confusions, offer diagnostic helps, or provide answers. A sample Open SWMM post and two replies [25] are shown in Figure 4 where the concern regarding the number of threads that the user would choose to run SWMM was communicated.

We adapt *association rule mining* [2] for discovering patterns in the user forum data, which represents a step toward automating the construction of metamorphic relations [16]. Association rule mining was originally developed to identify products in large-scale transaction data recorded in supermarkets. For example, an association rule {diaper} ⇒ {beer} would indicate that customers who purchase diapers are also likely to purchase beer. In this example, diaper and beer are called *antecedent* and *consequent* respectively. Apriori [2] is among the most well-known algorithms to mine associate rules from a database containing various transactions (e.g., collections of items bought by customers).

It is therefore critical to define *transactions* in the context of user forums for algorithms like Apriori to work. As different users have different viewpoints and use different vocabularies, their posts shall be treated as different transactions. In addition, posts at different times reflect the user's evolving views, possibly influenced by the thread of discussions. Based on these observations, we deem a distinct forum user's post at a single time as a transaction, much like a customer's purchase at a given time being considered as a transaction in market basket analysis. As a result, Figure 4 contains three transactions.

The raw posts shown in Figure 4, however, must be processed to make the transactions amenable to association rule mining. Algorithm 1 of Figure 5 shows our procedure to generate I/O associations. The pre-processing (lines 1–16) is

Q **User** 15-Feb-2018

I am simulating 3 urban catchments (totally 15.7 Km2). There are 273,000 stormwater pipes and the period of simulation is 18 years (rainfall data). When I only run the area for 2 months period (6 minutes time step) program shows up to 150 hours for running.

...

A **Commenter**

...

Finaly, are you using the maximum number of threads?

Q **User**

Many thanks for your useful comments. Yes, I have 237K pipes. I changed the threads from 1 to 4 and the simulation time has changed significantly. Now I am running for 10 years rainfall within 10 minutes.

...

Fig. 4. Sample Open SWMM post and two replies.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_29

```
Algorithm 1: Generate {I} ⇒ {O} Association Rules
   Input: a set of variables V manually identified from
          user manual, a user forum U
   Output: an unordered list L of input-output associations
1  Pre-processing:
2  U ← to_lower_case(U);
3  for (each v ∈ V) ∧ (each u ∈ U) do
4      VN ← {v.name} ∪ {v.alias};
5      while VN ≠ ∅ do
6          ln ← longest_name(VN);
7          if string_match(u, ln) ≥ δ then
8              substitute u with ln in U;
9              break;
10         else
11             VN ← VN \ {ln};
12         end
13     end
14 end
15 U ← preserve_and_unify_variable_name(U);
16 T ← split(U);
17 Processing:
18 L ← Apriori_algorithm(remove_punctuation(T), min_support);
19 Post-processing:
20 L ← (L.antecedent ∩ V.input) ∧ (L.consequent ∩ V.output);
```

Fig. 5. Mining association rules from a user forum.

tailored for user forum data U. Upon converting U into lower cases, Algorithm 1 sorts each variable's name identified from the user manual V based on length. If a term $u \in U$ matches the longest variable name $ln \in V$, then a match is found and u is replaced by ln (lines 8–9); otherwise, the next longest variable name is examined (line 11 continued with the while-loop back to line 5). This ensures that "wet weather time step" is recognized before "time step" is recognized.

We currently employ Levenshtein distance [10] and its fuzzywuzzy Python library (github.com/seatgeek/fuzzywuzzy) to implement *string_match* at line 7 of Figure 5, and the threshold δ=0.85 is determined heuristically by a small-scale pilot trial based on SWMM. Lines 15–16 show that pre-processing is completed with preserving and unifying the variable names in U, followed by splitting U into transactions. Once transactions are prepared, Algorithm 1 invokes Apriori to mine association rules where punctuations are removed from T. The post-processing of line 20 is to ensure that each rule's antecedent contains the input variable, and the consequent contains the output variable. We rank the mined association rules by Algorithm 1 via two metrics [2]: first with *support* that indicates how frequently the antecedent and consequence (i.e., the I/O variables) co-appear in the transactions, and then with *confidence* that determines the

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_29

relative amount of the given consequence across all alternatives for a given antecedent. Table 2 lists the ten top-ranked association rules mined from the Open SWMM posts and their support and confidence values. More complete results of association rule mining can be found in our online data [23].

Table 2. I/O association rules mined from the Open SWMM posts based on a total of 15,958 transactions.

rank	association rule	support	confidence
1	{upstream} ⇒ {flow}	0.029	0.533
2	{downstream} ⇒ {flow}	0.027	0.532
3	{weir} ⇒ {flow}	0.022	0.543
4	{rain barrel} ⇒ {runoff}	0.018	0.518
5	{surcharge} ⇒ {flow}	0.011	0.539
6	{surface area} ⇒ {storage}	0.010	0.645
7	{previous area} ⇒ {total precipitation}	0.009	0.593
8	{depression storage} ⇒ {infiltration}	0.008	0.554
9	{wet step} ⇒ {runoff}	0.008	0.506
10	{shape curve} ⇒ {runoff}	0.006	0.933

3.3 Comparing Ranked Lists

The inputs and outputs identified from the user manual (Section 3.1) represent comprehensive yet static information, whereas the association rules mined from the actual software usage data of a user forum (Section 3.2) help uncover the dynamic regularities of inputs and outputs. To compare these two ranked lists of I/O associations, we adopt Kendall's τ which is a correlation measure for ordinal data [9]. The τ value ranges from -1 to 1 where values close to 1 indicate strong agreement between two rankings and values close to -1 indicate strong disagreement. We use the SciPy Python library [28] to calculate τ, and the `scipy.stats.kendalltau()` function implements the following measure:

$$\tau = \frac{P - Q}{\sqrt{(P + Q + T) * (P + Q + U)}} \tag{1}$$

where P is the number of concordant pairs, Q is the number of discordant pairs, T is the number of ties only in the first ranking, and U is the number of ties only in the second ranking.

In our analysis, we first identify the *overlapped* I/O pairs from two ranked lists, and then compute Kendall's τ for only the overlapped pairs. Figure 6 illustrates our calculation of two ranked lists, A and B, both having four pairs. However, only three pairs are shared which we keep in A' and B'. The ranking of the three remaining pairs is preserved from the original list. In A' and B', "<weir, flow>" and "<area, storage>" are concordant, because the former is ranked higher than the latter in both lists. Similarly, "<aquifer, storage>" and "<area, storage>" are also concordant. The discordant comes from

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_29

“<weir, flow>” and “<aquifer, storage>” because their relative rankings are different in A’ and B’. Using equation (1), the Kendall’s τ between A’ and B’ is: $\frac{2-1}{\sqrt{(2+1+0)*(2+1+0)}}$=0.33.

ranked list A
<weir, flow>
<area, flow>
<aquifer, storage>
<area, storage>

ranked list B
<aquifer, storage>
<weir, flow>
<outlet, flow>
<area, storage>

overlap $\Longrightarrow$

ranked list A’
<weir, flow>
<aquifer, storage>
<area, storage>

ranked list B’
<aquifer, storage>
<weir, flow>
<area, storage>

Fig. 6. Illustration of selecting the overlapped pairs and then calculating Kendall’s τ.

The results of comparing the 200 top-ranked I/O associations are shown in Figure 7. The number of overlapped I/O pairs increases in a linear fashion, and approximately a quarter (e.g., 40 out of 160) I/O variables are associated in both SWMM’s user manual and in the Open SWMM forum. Among the overlapped I/O associations, the Kendall’s τ correlation remains positive. This shows that the concordant pairs outnumber the discordant ones, which implies the degree of I/O associations is reasonably consistent between the user manual produced by the scientific software development team and the forum posts among the end users themselves. While we will make some qualitative observations of concordant and discordant I/O variables in Section 4, we next discuss some of the important aspects of our study that one shall take into account when interpreting our findings.

3.4 Threats to Validity

A threat to construct validity is how we define the degree of association between an input variable and an output variable. In particular, we use different measures for the two different data sources. As the user manual is written by somebody who is familiar with the scientific software, we quantify the I/O associations based on how coupled the two variables are within the textual part and the tabular part. From the thousands of end users’ posts, we mine association rules, aiming to discover: “Forum users who mentioned an input variable also mentioned an output variable”. We believe such measures account for the static and authoritative natures of the user manual, and the dynamic and idiosyncratic natures of the user forum.

An internal validity threat is our manual identification of SWMM’s I/O variables from the user manual. Although an almost perfect inter-rater agreement (Cohen’s κ=0.87) was achieved on a randomly chosen sample, our manual effort may have false positives and false negatives. Another threat relates to the parameter values that we chose in association rule mining: δ=0.85 (line 7 of Algorithm 1) and $min\_support$=3 (line 18 of Algorithm 1). The former is determined by a small-scale SWMM pilot trial, and the latter is informed by a prior association rule study in software engineering [33].

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_29

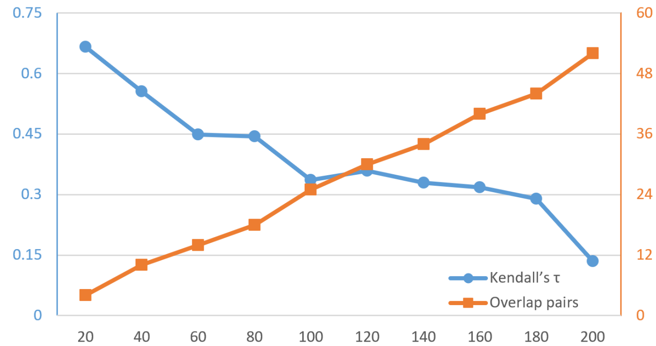


Fig. 7. Kendall's τ and the size of overlapped pairs between the I/O associations from the user manual (cf. Table 1) and the I/O associations from the user forum (cf. Table 2): x-axis represents the number of top-ranked I/O associations, left y-axis represents Kendall's τ values, and right y-axis represents the number of overlapped I/O pairs on which Kendall's τ is computed.

Several factors affect our study's external validity. Our results may not generalize to other user forums of SWMM and other scientific software systems. As for conclusion validity and reliability, we believe we would obtain the same results if we repeated the study. In fact, we publish all our analysis data in the institution's digital preservation repository [23] to facilitate reproducibility.

4 Discussion

While the Kendall's τ of Figure 7 shows positive correlations, we share some observations of the I/O pairs in the two ranked lists. A few I/O variables are ranked high in both lists, e.g., <rain barrel, runoff> is number one in Table 1 and number four in Table 2. We observe that the input variables are often necessary and yet end users may encounter some barriers of setting up the proper values. SWMM's user manual provides prototypical values, e.g., "...single family home rain barrels range in height from 24 to 36 inches (600 to 900 mm)" [26]. Another necessary and oftentimes misused input variable is "date" which the user manual specifies the permissible formats. However, different countries have different date conventions, making the concrete values from the user forum valuable for metamorphic testing, especially for selecting source test cases (cf. Figure 1).

Some I/O variables have associations stronger in the user forum than in the user manual. For instance, <shape curve, runoff> ranks tenth in Table 2 and 6055th in the user manual results. A closer look shows that the use of "shape curve" in the implementation became deprecated after version 5.0.015[1], and the variable "storage curve" should have been used. This indicates that association

[1] https://www.epa.gov/sites/production/files/2020-03/epaswmm5_updates.txt Last accessed: April 2021.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_29

Table 3. Comparing User Manual and User Forum of Scientific Software

	User Manual	**User Forum**
who & *why*	written by the development team	organized for & by end users
	focusing on the specific software	software playing a part in meeting goals
	usage norms of the software	idiosyncratic uses of the software
what & *how*	features & capabilities of the software	questions & dissatisfactions of software
	comprehensive intro to I/O	attentions paid to partial I/O
	prototypical I/O demos	actual I/O values
	updated periodically & authoratively	growing continuously & organically

rules mined from user forum posts may suggest problematic, and even deprecated variables. As a result, the user manual of scientific software shall be updated to better stay in sync with the evolution of the implementation.

Not only are some higher-ranked I/O associations from the user forum indicative of deprecation, they also reveal frequently used features of the scientific software. For example, <snowmelt, runoff> is the twelfth-ranked association rule mined from Open SWMM, but ranks 98th in the user manual's results. This shows that the user manual's descriptions tend to be comprehensive, making core parameters like "snowmelt" less prominent. In contrast, end users commonly discuss the important variables, as "Snowmelt parameters are climatic variables that apply across the entire study area when simulating snowfall and snowmelt" [26]. Interestingly, the association rules mined from the forum posts can depict the simulation capabilities used, potentially suggesting requirements and their evolution of the scientific software [11].

5 Conclusions

I/O associations are integral to metamorphic testing which has helped to address some scientific software testing challenges [8]. This paper reports our analysis of the user manual and user forum of EPA's SWMM in order to quantify I/O associations. Our results show a positive correlation of the identified I/O pairs, and further reveal the differences between the two data sources. Table 3 highlights the complementary aspects, which could assist in choosing the proper data to support scientific software's metamorphic testing, requirements engineering [11], software traceability [20], and other tasks.

Our future work includes developing automated and accurate ways to classify I/O variables, exploring associations beyond a single input variable and a single output variable, and instrumenting metamorphic testing with source test cases from the user manual and user forum. Our goal is to better support scientists in improving testing practices and software quality.

Acknowledgments. *We thank the EPA SWMM team, especially Michelle Simon, for the research collaborations. Funding is provided in part by the U.S. National Science Foundation Critical Resilient Interdependent Infrastructure Sys-*

tems and Processes (CRISP 2.0) Award to Dr. Omar I. Abdul-Aziz (NSF CMMI Award #1832680).

References

1. O. I. Abdul-Aziz and S. Al-Amin. Climate, land use and hydrologic sensitivities of stormwater quantity and quality in a complex coastal-urban watershed. *Urban Water Journal*, 13(3): 302–320, 2016.
2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *International Conference on Very Large Data Bases*, pages 487–499, 1994.
3. S. Al-Amin and O. I. Abdul-Aziz. Challenges in mechanistic and empirical modeling of stormwater: review and perspectives. *Irrigation and Drainage*, 62(S2): 20–28, 2013.
4. T. Y. Chen, J. Feng, and T. H. Tse. Metamorphic testing of programs on partial differential equations: a case study. In *International Computer Software and Applications Conference*, pages 327–333, 2002.
5. J. L. Fleiss and J. Cohen. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement*, 33(3): 613–619, 1973.
6. E. Huq and O. I. Abdul-Aziz. Climate and land cover change impacts on stormwater runoff in large-scale coastal-urban environments. *Science of The Total Environment*, 778: 146017, 2021.
7. S. Kamble, X. Jin, N. Niu, and M. Simon. A novel coupling pattern in computational science and engineering software. In *International Workshop on Software Engineering for Science*, pages 9–12, 2017.
8. U. Kanewala and T. Y. Chen. Metamorphic testing: a simple yet effective approach for testing scientific software. *Computing in Science & Engineering*, 21(1): 66–72, 2019.
9. M. G. Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3): 239–251, 1945.
10. V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Dokladay*, 10(8): 707–710, 1966.
11. Y. Li, E. Guzman, K. Tsiamoura, F. Schneider, and B. Bruegge. Automated requirements extraction for scientific software. In *International Conference on Computational Science*, pages 582–591, 2015.
12. X. Lin, M. Simon, and N. Niu. Exploratory metamorphic testing for scientific software. *Computing in Science and Engineering*, 22(2): 78–87, 2020.
13. X. Lin, M. Simon, and N. Niu. Hierarchical metamorphic relations for testing scientific software. In *International Workshop on Software Engineering for Science*, pages 1–8, 2018.
14. X. Lin, M. Simon, and N. Niu. Releasing scientific software in GitHub: a case study on SWMM2PEST. In *International Workshop on Software Engineering for Science*, pages 47–50, 2019.
15. X. Lin, M. Simon, and N. Niu. Scientific software testing goes serverless: creating and invoking metamorphic functions. *IEEE Software*, 38(1): 61–67, 2021.
16. X. Lin, M. Simon, Z. Peng, and N. Niu. Discovering metamorphic relations for scientific software from user forums. *Computing in Science and Engineering*, 23(2): 65–72, 2021.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_29

17. C. Murphy, G. E. Kaiser, L. Hu, and L. Wu. Properties of machine learning applications for use in metamorphic testing. In *International Conference on Software Engineering & Knowledge Engineering*, pages 867–872, 2008.
18. L. Nguyen-Hoan, S. Flint, and R. Sankaranarayana. A survey of scientific software development. In *International Symposium on Empirical Software Engineering and Measurement*, pages 1–10, 2010.
19. N. Niu, A. Koshoffer, L. Newman, C. Khatwani, C. Samarasinghe, and J. Savolainen. Advancing repeated research in requirements engineering: a theoretical replication of viewpoint merging. In *International Requirements Engineering Conference*, pages 186–195, 2016.
20. N. Niu, W. Wang, and A. Gupta. Gray links in the use of requirements traceability. In *International Symposium on Foundations of Software Engineering*, pages 384–395, 2016.
21. Open SWMM. SWMM Knowledge Base. https://www.openswmm.org Last accessed: April 2021.
22. A. Pawlik, J. Segal, and M. Petre. Documentation practices in scientific software development. In *International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 113–119, 2012.
23. Z. Peng, X. Lin, and N. Niu. Data of SWMM I/O Associations. https://doi.org/10.7945/0mn5-p763 Last accessed: April 2021.
24. Z. Peng, X. Lin, and N. Niu. Unit tests of scientific software: a study on SWMM. In *International Conference on Computational Science*, pages 413–427, 2020.
25. S. Rashetnia. Long simulation time for very large models. https://www.openswmm.org/Topic/11289/long-simulation-time-for-very-large-models Last accessed: April 2021.
26. L. A. Rossman. Storm Water Management Model User's Manual Version 5.1. https://www.epa.gov/sites/production/files/2019-02/documents/epaswmm5_1_manual_master_8-2-15.pdf Last accessed: April 2021.
27. R. Sanders and D. Kelly. Dealing with risk in scientific software development. *IEEE Software*, 25(4): 21–28, 2008.
28. SciPy. Scientific Computing Tools for Python. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kendalltau.html Last accessed: April 2021.
29. United States Environmental Protection Agency. Agency-wide Quality System Documents. https://www.epa.gov/quality/agency-wide-quality-system-documents Last accessed: April 2021.
30. United States Environmental Protection Agency. Storm Water Management Model (SWMM). https://www.epa.gov/water-research/storm-water-management-model-swmm Last accessed: April 2021.
31. United States Geological Survey. Review and Approval of Scientific Software for Release (IM OSQI 2019-01). https://www.usgs.gov/about/organization/science-support/survey-manual/im-osqi-2019-01-review-and-approval-scientific Last accessed: April 2021.
32. S. A. Vilkomir, W. T. Swain, J. H. Poore, and K. T. Clarno. Modeling input space for testing scientific computational software: a case study. In *International Conference on Computational Science*, pages 291–300, 2008.
33. W. Wang, N. Niu, M. Alenazi, J. Savolainen, Z. Niu, J-R. C. Cheng, and L. D. Xu. Complementarity in requirements tracing. *IEEE Transactions on Cybernetics*, 50(4): 1395–1404, 2020.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_29