SMMM: A Metric Based Framework To Evaluate The Scalability Of Multiple Model Methodologies

Akhilesh Bajaj
The Heinz School
Carnegie Mellon University
akhilesh@andrew.cmu.edu

Abstract:
Multiple Model Methodologies (MMM) have become ubiquitous in the area of conceptual modeling. Thus, the Unified Modeling Language (UML) is a popular MMM in software engineering, while MMs are also used in enterprise modeling. Over the last few years, the size of problem domains being modeled by MMs has also grown. Thus, software is now bigger, and enterprises are significantly larger in scale than the problem domains modeled when building the legacy systems. These trends in the area of conceptual modeling raise an important question about the scalability of MMs, as they are applied to domains of increasing size. In this work, we present a comprehensive, metric based framework called SMMM (Scalability of MMM). SMMM assumes that the only obstacle to the scalability of an MMM is the complexity that users face when using the MMM to create models of the underlying reality, as this reality grows in size. SMMM extends previous work in the area of complexity measurement in the following ways. First, SMMM is comprehensive, and yet parsimonious. Second, metrics in earlier works have made explicit assumptions about the relative cognitive difficulties of modeling different categories of concepts. SMMM makes no assumptions about any concept being harder to model than another. Third, previous work on metric development has omitted the role of empirical work in understanding complexity. The SMMM framework explicitly recognizes the role of empirical work in evaluating cognitive difficulty. Fourth, SMMM measures both intra-model and inter-model complexity. Intra-model complexity values show which models in the MMM are the source of complexity. Inter-model complexity values measure the complexity by the interaction between different models in the MMM.

1. INTRODUCTION

Recently, there has been increased interest in using modeling methodologies which use multiple models, to offer different views to different players. Multiple models are being used in the area of software engineering e.g., the Unified Modeling language (UML) [Booch, Jacobson, and Rumbaugh, June 1997], as well as in enterprise modeling [Loucopoulos and Kavakli, 1999]. In both areas, the problem domains are increasing in size. Thus, the size of software systems is growing, and modeling an enterprise is a much larger problem than modeling an isolated system, as was done in legacy systems. The scalability of the methodologies being used to model a large problem domain, such as an enterprise or a large software, is becoming increasingly important [Burkhart, 1992]. In this work we propose a comprehensive, metrics based framework that can be used to measure the scalability of multiple model methodologies (MMM). We call this framework SMMM (Scalability of MMM). We propose both intra- and inter-model metrics for measuring scalability. We also illustrate the usage of this framework by evaluating an MMM consisting of the Entity Relationship Model (ERM) [Chen, 1976] and the Data Flow Diagram (DFD) model [Gane and Sarson, 1982].
In this work we assume that the only obstacle to the scalability of an MMM is the complexity that users face when using the MMM to create models of the underlying reality, as this reality grows in size. Hence, we focus on developing metrics to measure this complexity.

There is a growing body of work (e.g., [Bajaj and Krishnan, 1999; Brinkkemper et al., 1999; Moynihan, 1996; Rossi, 1996; Rossi and Brinkkemper, 1996]) that proposes metric and non-metric approaches to evaluating the complexity and/or usability of a single conceptual model. There has also been much less work that recognizes the complexity associated with using MMMs [Brinkkemper et al., 1994; Castellini, 1998]. The primary contribution of this work is to extend this past work on MMMs by proposing a framework that a) explicitly creates metrics for intra- and inter-model complexity, and b) recognizes and frames the role of empirical work in evaluating the complexity that arises from using an MMM.

To the best of our knowledge, previous work on evaluating MMMs makes no attempt to reconcile what can be learned from non-empirical metrics and what can only be learned from empirical studies. The SMMM framework represents a first step in recognizing the role of both non-empirical and empirical metrics in evaluating complexity. The SMMM framework presented here also differs significantly from previous work that uses meta-languages (e.g., [Rossi, 1996]) because it does not categorize the concepts found within models into different pre-determined categories, and then attempt to evaluate the complexity based on the number of concepts within each category.

The rest of this paper is organized as follows. In section 2, we provide the research context of our work. In section 3, we present the scalability metrics that form SMMM. In section 4, we illustrate the usage of SMMM by presenting a detailed analysis of an MMM. The conclusion and directions for future research are presented in section 5.

2. PREVIOUS WORK

The use of MMMs to analyze and design information systems (IS) predates the enterprise modeling area by several years. Thus, Sakai, [1984] proposed combining the entity ERM with behavior diagrams (using Petri nets) to capture data and processes. Shoval, [1991] proposed that the functional, process and data views are needed to model a system. He also proposed an MMM that explicitly linked some concepts of each model in the MMM, resulting in an integrated methodology. Another integrated methodology was proposed by [Wieringa, 1995], who also modeled the functional, process and data aspects. This methodology used transactions as a pivot to explicitly link concepts in each of the models. It has become widely accepted in the literature that at the very least, the data and process aspects of a system need to be modeled, and usually MMMs are proposed to accomplish this.

In recent years, enterprise modeling has become increasingly important in the conceptual modeling literature. Several methodologies describing enterprise modeling are summarized in [Petrie, 1992]. As an example of an enterprise modeling methodology, Loucopoulos and Kavakli [1999] proposed that, in addition to the traditional data and process models that have been used to model what an enterprise ‘does’, it is also important to capture the goals of the enterprise, the rationale behind these goals, the players in the enterprise, and the roles played by these players. They proposed an enterprise knowledge metamodel, that captures these concepts in detail. The metamodel defines three submodels: a goal submodel, a process submodel and a component submodal, thus creating an MMM.
In the area of software modeling, the recent emergence of the UML as an informal standard signifies that MMMs are likely to be increasingly used in software engineering also. Most commercially available Computer Aided Software Engineering (CASE) tools use multiple methodologies. A representative example of these CASE tools is Designer/2000 by Oracle Corp., which uses seven different models for process modeling, systems modeling and systems design [Page, 1996]. The recently emerging UML standard uses nine different diagrams [Booch, Jacobson, and Rumbaugh, June 1997].

Based on this review of literature, we conclude that MMMs are ubiquitous in the area of conceptual modeling today.

Several authors have raised concerns about the complexities of using MMMs. [Marttiin et al., 1995] propose that a “lack of mechanisms for integrating sets of methods while maintaining consistency between various models” is a major weakness in current CASE tools.

Much of the concerns regarding the complexity of MMMS have centered around the design of the software (or tools) as opposed to the inherent models in the MMM. Thus, Brinkkemper et al., [1994] recognize two main problems when dealing with the complexities of upper CASE tools (used in systems analysis): the complexities within diagrams and the relationships between diagrams of different types. They suggest that complexities within diagrams can be reduced by viewing certain parts of the schemas in isolation at a time (viewing) and creating top down schemas so that schemas may be viewed at different levels of abstraction (hiding). Complexities between diagrams can be reduced by a technique called transparency. In [Brinkkemper et al., 1994], transparency means the ability of the tool to recognize that the same concept is being represented in different diagrams, and to automatically link these concepts together. They define transparency on a scale of 0 – 3. A transparency value of 0 refers to CASE tools where the users are forces to keep everything consistent manually. Transparency values of 1 and 2 imply that the CASE tool recognizes the common concepts in different diagrams, and automatically updates all diagrams. A transparency value of 3 implies that the user can specify links between objects in different diagrams, and the CASE tool automatically updates for consistency. In a survey of eight CASE tools, they found that most tools have some support for transparency, but it is usually less than 2.

Another widely recognized need for CASE software is the need to have consistent data in the CASE repository. Several authors (e.g., [Brown and Carney, 1993; Kubalski et al., 1993; Rader, Morris, and Brown, 1993]) have also recognized that creating more perspectives leads to greater integration requirements on the software tool.

SMMM differs from these past works in that it focuses only on complexities associated with using the models underlying the MMM, not in using the (software) tools that are used to create model schemas.

In the area of evaluating the complexity of models, most of the previous work proposes metrics that only account for intra-model complexity. Marcos, Cervera, and Fernandez [1999] defined a size metric (the number of concepts in the model) and the relationships metric (the number of relationship pairs between concepts within a model). They analyzed five models, and concluded that the ERM is the least complex, while UML is the most complex. Rossi [1996] also defined metrics to evaluate the complexity within a model. These metrics measure the number of objects, the number of relationships between the objects, and the total number of properties (of objects and relationships) in a model. He also proposed aggregate metrics, one of which compares the
degree to which an object is described by its properties versus its relationships. To the best of our knowledge, the only work that examines inter-model complexity is [Castellini, 1998], who proposed a charts of concepts approach to evaluate the size, complexity and the errors in definition of an MMM. The metrics in this approach involve grouping the concepts in an MMM into charts, and then evaluating the number of concepts per group, as well as in total. The complexity is evaluated by measuring the overlap in the charts (the number of charts in which a concept occurs), and there are several metrics that evaluate the definition dependencies between concepts. The errors in an MMM include the presence of circular definition dependencies, where a concept ends up defining itself.

Based on this survey we conclude that first, previous research on model complexity has focused primarily on intra-model complexity, though there is now increasing recognition of the idea that MMMs present a source of both intra- and inter-model complexity. Second, it is recognized that both the number of concepts in an MMM as well as the overlap / relationships between concepts are important sources of this complexity. However, most work has focussed on studying the links between concepts within a model, while ignoring the possibility of links across models. Third, to the best of our knowledge, there is a lack of metrics that measure the inherent cognitive difficulty associated with using a concept in a model. Fourth, almost all metrics that have been proposed up to now are non-empirical in nature. Even though it seems intuitive that empirical studies are needed to evaluate complexity, there has been little work done to frame the roles that empirical and non-empirical metrics must play together, to allow a more complete and accurate picture of complexity.

We next propose SMMM, a metrics based framework that a) explicitly proposes metrics for intra- and inter-model complexity and b) recognizes the role of both non empirical and empirical metrics in evaluating the complexity of MMMs.

3. SMMM: A METRICS BASED FRAMEWORK FOR EVALUATING THE SCALABILITY OF MMMs

3.1 Metrics to Measure Intra-Model Complexity
There appear to be two sources of intra-model complexity. The first is the size of the model itself, which has been measured in many ways in previous work. Thus, as the number of concepts in a model increases, it becomes more complex to use the model. The second source is the cognitive load that results from actually using each concept. Thus, two models with the same number of concepts may differ in complexity, if it is cognitively harder to use the concepts in one model. As a concrete example, the cardinality concept in the ERM may be harder to use than the entity set concept.

We define the concepts in a model to be all the elements of that model. For example, in the ERM, the concepts would consist of entity sets, relationship sets, attributes of entity sets, attributes of relationship sets, cardinality, and primary key of entity sets.

The metric used to measure the size of the model is the Metric Size of Model (MSM):

$$\text{MSM} = \sum \text{concept},$$

where the summation is across the concepts $i$ in the model.

There is some recognition in previous work that this metric alone is not sufficient to give the true measure of intra model complexity. That is why several metrics in earlier studies (see section 2)
attempt to categorize concepts by using a meta language (for example into objects, properties, relationships and roles (OPRR) [Rossi, 1996]). Metrics that measure complexity by examining the number of concepts in different predetermined categories assume innately that some predetermined categories are cognitively harder to use than others. For example, Rossi [1996] assumed that relationships are harder to use than properties, in the OPRR meta language. However, this assumption may not always be true. Thus, the IS-A concept in the extended ERM (which would be a relationship in OPRR) may be easier to use than, say, the primary keys of relationship sets in the extended ERM (which would be a property in OPRR).

In SMMM, we do not predetermine categories into which all models’ concepts must fall. Rather than make a priori assumptions (about all models) about which categories of concepts may be harder to use than others, we represent the complexity of using a concept by two metrics. The first metric is non-empirical and measures the ambiguity of a concept. We define the ambiguity \( A_{ij} \) between two concepts \( i \) and \( j \) (in the same model) as true if the concept pair can be used to represent the same piece of reality, and we say that \( \langle i, j \rangle \) is an ambiguous pair. For example, in the ERM, the same reality can be represented as an entity set or as a relationship set. The more ambiguous a concept, the more the cognitive load on the user, which makes the model less scaleable, as mentioned in section 1. We only count each ambiguous pair of concepts once, to evaluate ambiguity. So \( \langle \text{entity set, relationship set} \rangle \) and \( \langle \text{relationship set, entity set} \rangle \) will be counted only once. For \( N \) concepts, there are a total of \( \frac{N(N-1)}{2} \) possible pairs.

We also note that these pairs must obey the law of transitive closure. More formally,

\[ \langle i, j \rangle \land \langle j, k \rangle \rightarrow \langle i, k \rangle \]

The metric used to compute the average ambiguity of the model is the Metric Ambiguity Concepts (MAC) metric:

\[
MAC = \frac{\left( \sum A_{ij} \right) \times 100}{N(N-1)/2}
\]

where the summation is across the concepts \( i \) and \( j \) in the model, such that \( A_{ij} \neq A_{ji} \).

The second metric we use to represent complexity recognizes that some concepts are harder to use than others. For example, the cardinality concept in the ERM is harder to use than the entity set concept.

We contend that empirical evaluation is needed to evaluate this aspect of complexity. Thus it appears to be very difficult to determine a priori, without empirical testing, that it is harder to map from the underlying reality to one concept than it is to another. To recognize this, we include in SMMM an empirical metric we call the Metric for Mapping Complexity (MMC). There are several ways to define MMC, depending on the empirical method used. One possible definition of this metric is to present cases of increasing complexity to subjects, and measure the number of errors they make for a concept when creating model schemas. A second possibility is to prepare a Likert scale questionnaire where subjects self report their perceptions about the values of MMC, though user perception may not be correlated with actual performance. The proposal of a rigorous methodology for evaluating the MMC metric is part of our future work.

To summarize, we have proposed two non-empirical metrics (MSM and MAC), and one empirical metric (MMC) to evaluate intra-model complexity. We next present metrics for evaluating inter model complexity.

---

1 We note that our aim is not to single any work out for critique, but rather to illustrate a trend in previous work by means of a representative example.
3.2 Metrics to measure Inter-Model Complexity

There are two sources of inter-model complexity. The first source, as is recognized by [Castellini, 1998], is the overlap of concepts across models. For example, in UML, objects occur in both object diagrams as well as in sequence diagrams. Overlaps adds to complexity because the modeler has to keep consistent the occurrences of the concept across models. Even with the assistance of a software tool, ultimately the modeler has to provide input as to which concept occurrence in a model is equivalent to an overlapping concept occurrence in another model. The second source of complexity is that certain concepts in different models are involved in consistency relationships. We give two brief examples of consistency relationships. In an MMM that uses structure charts [Yamamoto, 1995] and the DFD model, a primitive process in the DFD will occur as a module on the structure chart. Hence a consistency relationship exists between primitive processes (in the DFD) and modules (in the structure chart model). As a second example, in a methodology that uses the DFD model and the ERM, the dataflow concept (in DFD) is involved in the consistency relationship with all the concepts in the ERM, the relationship being that the components of the dataflow have to exist somewhere in the ERM schema. We now define the two metrics to evaluate these two sources of complexity.

We define the overlap $O_i$ of a concept $i$ to be the (number of models – 1) in an MMM in which the concept appears. The maximum possible overlap of a concept $i$ in an MMM is $M-1$, where $M$ is the total number of models in the MMM.

The metric used to compute the mean overlap for an MMM is the Metric for Overlap (MO):

$$MO = \frac{\sum O_i \times 100}{N \times (M-1)}$$

where the summation is across all the concepts $i$ in all the models in the MMM.

The denominator in the formula for MO represents the maximum possible overlap of all $N$ concepts in the MMM, with each concept having a maximum possible overlap of $M – 1$.

For the second metric, we note that the relationship between the concepts can be any consistency relationship. We note that the consistency relationship is symmetric between models, so we only evaluate the relationship once between two models. In order to do this, we order the models arbitrarily so each one gets a number from 1 . . . $M$. We denote the total number of concepts in model $p$ as $N_p$.

Let $N_{\geq p}$ be the total number of concepts in models whose numbers are greater then $p$.

$$N_{\geq p} = \prod_{j=p+1}^{M} N_j$$

The maximum possible number of concepts with which a model $p$ can have a consistency relationship is $N_p \times N_{\geq p}$. This is because each concept in model $p$ can have a relationship with $N_{\geq p}$ concepts.

Let $R_i$ be the total number of concepts in models greater than $p$ with which the $i$ th concept in model $p$ has a consistency relationship. Then the total number of concepts with which model $p$ has a consistency relationship is
\[ R_p = \frac{N_p}{\prod_{i=1}^{M-1} R_i} \]

The ratio \( R_p / N_p \) measures what fraction of possible consistency relationships exist for the model \( p \), with models whose number is greater than \( p \). Adding this across all models up to \( M - 1 \) and dividing by \( M - 1 \) gives us the fraction of possible consistency relationships across all models. Since the relationships are symmetric, the last model (with number \( M \)) is not considered, since it has already been linked to every other model.

The metric used to compute the mean number of relationships across the MMM is the Metric for Consistency Relationships (MCR):

\[
\text{MCR} = \left( \frac{\sum_{p=1}^{M-1} \frac{R_p}{N_p} * N_p}{M - 1} \right) * 100
\]

The properties of the four non-empirical metrics in SMMM are summarized in table 1. We mention the range of values for each metric, the direction of correlation with complexity, and whether the metric can be used to compare models and MMMs of different sizes.

<table>
<thead>
<tr>
<th>Metric name</th>
<th>Range of Values</th>
<th>Direction of Correlation with Complexity</th>
<th>Comparable across models or MMMs of different sizes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSM</td>
<td>0 – infinity</td>
<td>Positive</td>
<td>Yes</td>
</tr>
<tr>
<td>MAC</td>
<td>0 – 100</td>
<td>Positive</td>
<td>Yes</td>
</tr>
<tr>
<td>MO</td>
<td>0 – 100</td>
<td>Positive</td>
<td>Yes</td>
</tr>
<tr>
<td>MCR</td>
<td>0 – 100</td>
<td>Positive</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1. Summary Information On Non-Empirical Metrics In The SMMM Framework.

We next illustrate usage of the SMMM framework by evaluating an MMM that uses a ERM model and the DFD model.

4. USING SMMM TO EVALUATE THE SCALABILITY OF AN MMM CONSISTING OF THE ERM AND DFD MODELS

We first evaluate the intra-model metrics. The ERM we use here as an example, consists of the following concepts: entity set, relationship set, attribute of entity set, attribute of relationship set, cardinality, and primary key of entity set. Thus, the MSM value for ERM is 6. The concepts in the DFD model are external entity, process, decomposition link (connecting a process and its parent), primitive tag (designating a primitive process), mini-spec (verbal description of a primitive processes), dataflow and data store. Thus the MSM value for the DFD diagram is 7. We note that we are only using the concepts here as an example, there may be other concepts that may be added, such as the degree of a relationship and aggregate relationships.

\(^2\) The empirical metric MMC is not listed, since it’s properties are not defined in this work. MMC has been included in this work to complete the SMMM framework, and delineate the role of empirical and non-empirical metrics.
In order to compute MAC, we need to list the concept pairs that can potentially be used to model the same reality. We note that the preparation of such a list requires previous experience on part of the researcher on how to use the models. The concept pairs list for the ERM is:
<entity set, relationship set>,
<primary key of entity set, attribute of entity set>,
<attribute of entity set, attribute of relationship set>
<primary key of entity set, attribute of relationship set>

Thus the MAC value of the ERM = (4/15)*100= 26.67.

The concept pairs list for the DFD model is:
<external entity, process>
<data flow, data store>

Thus the MAC value of the DFD model = (2/21)*100 = 9.52

In order to compute the MMC value, a suitable empirical methodology will need to be adopted. Section 3.1 has some preliminary suggestions for this. Based on the values of the non-empirical metrics alone, we observe that the ERM is a greater source of complexity than the DFD model. Thus, intra-model metrics can help us pinpoint the source of complexity in an MMM. We next compute the inter-model metrics.

Based on the listing above of all the concepts in the two models, we see that there is no overlap between the two models. Note that an overlap will occur if the same concept is present in more than one model, not if concepts are merely related to each other (this second part is captured in the MCR metric). Hence the MO value for the MMM we are considering = 0.

Next we compute the MCR metric. Let us associate the ERM with the number 1, and the DFD model with the number 2. The list of concepts in the ERM in a consistency relationship with the DFD model is shown below:
entity set <dataflow, data store>
relationship set <data flow, data store>
attribute of entity set <data flow, data store>
attribute of relationship set <data flow, data store>
primary key of entity set <data flow, data store>

In this case:
M = 2,
R = 2+2+2+2+2+0 = 10,
Np* Np = 6*7 = 42.
Therefore, MCR = (10/42/1)*100 = 23.81.

The listing of non-empirical metric values for the MMM is contained in table 2.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSM</td>
<td>ERM = 6, DFD = 7</td>
</tr>
<tr>
<td>MAC</td>
<td>ERM = 26.67, DFD = 9.52</td>
</tr>
<tr>
<td>MO</td>
<td>0</td>
</tr>
<tr>
<td>MCR</td>
<td>23.81</td>
</tr>
</tbody>
</table>

Table 2. Non-Empirical Metric Values For The MMM Consisting Of ERM And DFD
5. CONCLUSION

5.1 Contributions
In this work, we present a comprehensive framework called SMMM to measure the scalability of MMMs. The work makes several contributions. First, SMMM is comprehensive, and yet parsimonious. Second, metrics in earlier works have made explicit assumptions about the relative cognitive difficulties of modeling different categories of concepts. SMMM makes no assumptions about any concept being harder to model than another. Third, previous work on metric development has omitted the role of empirical work in understanding complexity. The SMMM framework explicitly recognizes the role of empirical work in evaluating cognitive difficulty. Fourth, SMMM measures both intra-model and inter-model complexity. Intra-model complexity values show which models in the MMM are the source of greater complexity. Inter-model complexity values measure the complexity by the interaction between different models in the MMM.

We do not consider intra-model consistency relationships in SMMM, though these have been considered in previous work (e.g., [Marcos, Cervera, and Fernandez, 1999]). The reason we do not include them is that the rationale for any model is to provide a cohesive view of reality, and thus there are always many consistency relationships between the concepts within a model. In our view, tight cohesiveness within a model does not increase complexity, whereas consistency relationships across models do. This concept is similar to the well understood concept of cohesion versus coupling in software engineering, where cohesion is not a source of complexity, but coupling is a source of complexity.

5.2 Future Work
An immediate extension of this work is to propose and validate an empirical methodology to evaluate the empirical metric MMC. While SMMM seeks to measure scalability (by measuring complexity) it does not measure other important properties of MMMs such as expressive power. Thus, it may well be possible that models that are more complex (and hence less scaleable) are also more expressive. An interesting extension of the SMMM framework will be to incorporate these other properties such as expressiveness within the framework. Currently, SMMM focuses only on complexity associated with the underlying models. It does not include metrics to measure the complexity associated with the software tools used to create MMM schemas, though this has been the focus of much past work. Such metrics can be incorporated into SMMM in order to get a more complete picture of complexity.

REFERENCES


