Journal of the International Society for the Systems Sciences | 68th Meeting of the International Society for the Systems Sciences
Wilby, J., Makar, J. and Friend, M. (eds)

APPLIED SYSTEMS SCIENCE TO HOLISTIC QUALITY ASSESSMENT METRICS FOR FORMAL METHODS-BASED MODELS

Haider Al-Shareefy
ECS, University of Southampton, Southampton, UK
hal1e20@soton.ac.uk
haideralshareefy@outlook.com

Dr. Stephen Wright
Department of Engineering, Mathematics and Design, University of West of England, Bristol, UK
Steve.Wright@uwe.ac.uk

Abstract
In MBSE, deciding on the superiority of one formal method-based specification over another can be challenging, especially when both fulfil the exact requirements in distinct manners. This paper delves into a methodology for distinguishing between formal methods, using Event-B notation as a primary example, based on inherently subjective quality aspects. Traditional complexity metrics used in software assessment do not align well with formal methods, prompting the need for a novel quality function that considers the impact on the reader's cognitive stress. The quality function incorporates classical axiomatic properties from theoretical mathematics and a bespoke complicatedness metric to evaluate model quality. Four critical properties—Consistency, Completeness, Independence, and Complicatedness—serve as the basis for this evaluation. Our approach suggests that if a formal specification appears visually complicated to someone with basic set-theory knowledge, it likely indicates a less helpful model regarding assurance. Additionally, the paper reconsiders Miller's magic number seven, proposing that understanding human cognitive limits and using magical numbers 1, 2, and 3 is essential for defining high-quality formal specifications. This innovative perspective underscores the need for extensive further research, as outlined in the discussions of future work.

Keywords

Formal Methods, Systems Thinking, Event-B, Model Based System Engineering, Miller's Rule, Complicatedness Function, Complexity Function

1 Introduction

As we were evaluating a formal specification for an aircraft landing gear case study, we realised that there is no specific scheme available for formal methods. Formal methods practitioners make their own judgment on the size and complexity level to be introduced as part of a solution without the guidance of some systemic metric that can keep track of how their formal design decisions impact the system as a whole. While formal methods are meant to eliminate ambiguity and check the validity of system requirements, the complexity of formal models can sometimes lead to confusion. For example, Exhibit 1 shows only one fragment of a larger statement; note how complex the chosen formalism is.

Exhibit 1. An illustration of what we would classify as a very complicated model (M´ery, 2014)

$$\texttt{act31} : general\_EV\_func :\in (1..3 \rightarrow POSITIONS) \times (1..3 \rightarrow A\_Switch) \times (1..3 \rightarrow (GEARS \rightarrow BOOL)) \times (1..3 \rightarrow (GEARS \rightarrow BOOL)) \times (1..3 \rightarrow GEAR\_ABSORBER) \times (1..3 \rightarrow (DOORS \rightarrow BOOL)) \times (1..3 \rightarrow (DOORS \rightarrow BOOL)) \times (1..3 \rightarrow BOOL) \rightarrow BOOL$$

$$\texttt{act32} : close\_EV\_func :\in (1..3 \rightarrow POSITIONS) \times (1..3 \rightarrow A\_Switch) \times (1..3 \rightarrow (GEARS \rightarrow BOOL)) \times (1..3 \rightarrow (GEARS \rightarrow BOOL)) \times (1..3 \rightarrow GEAR\_ABSORBER) \times (1..3 \rightarrow (DOORS \rightarrow BOOL)) \times (1..3 \rightarrow (DOORS \rightarrow BOOL)) \times (1..3 \rightarrow BOOL) \rightarrow BOOL$$

It can be noted how some practitioners choose to over-complicate formal specifications. Such a trend in synthesising formal specifications can have a counterproductive impact on addressing the ambiguity of specification. Therefore, this paper provides a metric for classifying and quantifying the quality of formal methods. This should assist developers in maintaining the simplest yet most informative form possible for describing a specification. Furthermore, quality quantification metrics can contribute to a cost estimation process for formal design development. To give an arbitrary example, evaluating working hours taken to produce a given complexity level. This paper is an extended and further improved version of an earlier research version submitted and accepted at the MEDI 2021 conference.

2 Literature Review

2.1 Complexity Measures

Complexity measures in software engineering should serve both descriptive and prescriptive purposes (Kearne, 1986). Descriptive measures evaluate the software's error-proneness, understandability, and testability. Prescriptive measures, on the other hand, provide specific steps to control the software (Ardito & al., 2020). For example, splitting complex modules into simpler ones or indicating the required testing for specific modules. In addition, good complexity metrics should be predictive and correlate with something interesting (Tashtoush & al., 2014). Also, they should have a consistent meaning regardless of the programming language used. The most basic complexity measure is the number of lines of code. However, this method is not ideal for open reengineering since the source code's programming language, coding style, and textual formatting influence it (Hao & al., 2023). A better complexity measure is cyclomatic complexity, which quantifies the amount of decision logic in a software function. This method, also known as McCabe complexity, is a more suitable application of the open reengineering criterion. Examples of current complexity measures (Watson, 1996):

- Cyclomatic complexity v(G) : Amount of decision logic.
- Essential complexity ev(G): Amount of poorly- structured logic.
- Module design complexity iv(G): Amount of logic involved with subroutine calls.
- Data complexity (SDV): Amount of logic involved with selected data references.

McCabe complexity is calculated using the number of decision points + 1(for structured programming). For example, suppose an algorithm with 5 decision nodes (if statements); then McCabe's complexity number for this algorithm would be 6. McCabe's complexity is also related to a method/procedure's testability but not a specific path's testability. McCabe's complexity does not consider the feasibility of paths (in terms of whether there are specific values for the variables that force the execution through them) (Hummel, 2014). Contemporary software metrics mainly describe the software but cannot describe the comprehensibility of the software parts. Comprehensibility of the software parts can be related to the same psychological factor that limits people’s ability to do mental manipulations of no more than 7 objects simultaneously (Miller, 1956). The Halstead metric (Halstead, 1977) suggests several measures that draw upon a program's various operators and operands. Neither McCabe’s cyclomatic complexity metric nor Halstead’s proposal explicitly draws inspiration from the psychology of programming nor takes into consideration human factors.

2.2 Theoretical Mathematics: Axiomatic Systems (AS)

An axiomatic system is a list of a predetermined set of statements that are accepted as truths with no need for proof. A very good example of AS is Euclidean Geometry and axioms of the natural numbers. One of the reasons axiomatic systems are developed is the need to determine precisely which properties of systems can be deduced from self-evident properties (Lee, 1997). Any set of axioms can be described by three main properties: consistency, independence, and completeness (Barsamian, 2017). Typically, limiting axioms to include only statements simple enough to be intuitive and self-evident is sufficient for practical purposes. Therefore, the simplicity of proposed axioms can be identified as virtuous in axiomatic-invariant statements. David Hilbert, one of the most important mathematicians of the twentieth century, specified the simplicity of axioms as the fourth property of a good and well-founded axiomatic system (Corry, 2004).

2.3 Complexity and Complicatedness of Systems

Complexity may be defined as a function of the number of recognisable parts and the types of relations among them (Klir, 2001). It is also a relative concept in that the "complexity of an object is in the eyes of the observer" (Klir, 2001). Complexity may be further characterised by the interaction between the system and a human being (or a computer) for whom it may be complicated to comprehend or cope with system demands and performance outputs. From that, we distinguish the term "complicatedness" of complexity in that complexity as a term relates to the perceived phenomena itself, like the complexity of a train network system or the complexity of a formal specification.

Complicatedness relates to the impact of perceived complexity on the ability of a predictive observer to predict it and accurately predict and trace all perceived interactions and emergence from and to their contributing factors within the perceived complexity. In other words, how complicated a complexity is to a predictive observer. Phenomena may be complex, but not necessarily unpredictably complicated; they can be complicated but predictable, which relates to how confusing or apparent the complexity's complicatedness is with respect to a predictive observer. When we hear the term "how complex a problem or a system is", in reality, we are referring implicitly to "how complicated what we are observing for us to predict its behaviour so that we can predict and be ready to deal with its nature if we decide to establish a relationship with it". In that sense, the method to resolve a complexity is fundamentally about resolving its unpredictability by modelling its construct and inner interactions so that we understand what we do when surprises happen. In other words, solving complexity is about resolving its complicatedness. In Cynefin's sense, complexity is initially chaotically complicated, and then through a resolution methodology, we resolve the perceived unpredictable chaos to be predictable complicatedness. If you look again at Exhibit 1, your initial reaction when you saw it the first time is the complicatedness of the observation; when you look at it again, its complicatedness is somewhat less, but (depending on how much you know about formal systems) feels less complicated than your first. However, your perception of Exhibit 1 complicatedness may still be the same.

Therefore, complexity, with respect to a predictive observer, is at least (always) initially confusingly complicated (upon initial observation), and the observer must constantly battle to resolve their confusion of its complicatedness by modelling it in a way that becomes predictable and thus clear. Thus, the complexity's complicatedness can generally be in two states: *clear* or *confusing*. If a perceived complexity is confusing, it is a problem; if it is clear, it is a solved problem (notice we still used the word problem, not a solution, here because, due to universal entropy, it is safer to think of any solution as only temporary predictable situation, in fact, it is a potential problem). The general rule between complexity and complicatedness is that the more complex a phenomenon gets, the more likely it becomes confusing for an observer's thinking methods. We (as observers) seek to avoid confusion since it requires more energy and effort to make sense of a perceived complexity. Hence, our minds always seek to model abstract patterns in any observation.

Natural systems' complexity can be defined as the quantity of information required to describe them (Ashby, 1973). Considering such intuition, complicatedness can be defined as the quantity of information, steps taken, or cognitive energy (the physical energy spent to operate a cognitive process to

infer a prediction) required to determine abstract rules that explain systems' behaviour. It is also related to the time it takes a problem solver or an observer to understand any aspect of an observed complex (a system). If an observer takes more time to understand a system, the perceived complexity is more confusing than if it takes less time. From the latter intuition, we can model the relationship between complexity and complicatedness as follows (an example application is given in section 6.3):

1. **Scale of Perceived Complexity (S)**: This reflects the scale or magnitude of the perceived complexity. It can be quantitatively expressed (using any justified quantification method) as the number of components or interactions within the system (being perceived). The associated exponent *a* can capture a quantifiable aspect or a property about the complexity structure.
2. **Knowledge about the Perceived Complexity (K)**: K is about the observer's level of awareness. This factor accounts for what the observer already knows or is aware of the observed system or what known information exists in a language that can communicate information about the system to the observer, including familiarity with its nature, structure and behaviour rules. Prior knowledge can profoundly influence the observer's ability to interpret complexity and predict its past, present, and future. The exponent b can be related to more holistic quantifiable property regarding the observer's knowledge—for example, a level of expertise in the whole system the observation is part of.
3. **Latent Complicatedness (L_c)**: It can also be defined as latent confusion due to the complexity of the encompassing whole. This is the inherent difficulty based on the number and the Certainty of general rules governing the interactions observed within a perceived complexity. A volatile rule is an uncertain rule that governs the nature of interaction among variables whose correctness or validity may or may not hold the test of time, the expanding scale of complexity or the expanding size of the observation window. Some volatile rules remain unchanged longer than others. Lc term also can capture a weight representing a property related to the whole, where the observed view of the system is part of. Thus, this accounts for some aspects of holism in the measure. The more rules you introduce into a model of complexity, the more likely the complexity of the designed system will transmit confusion to another observer.
4. **Complicatedness** ζ: When we use *complicatedness, we refer to the likelihood of an observed complex becoming* confusing to some observer. It measures the likelihood that a predictive observer was, is or will be confused when perceiving a complexity (naturally existing or intendedly designed). We relate it to the Epistemic Uncertainty of the observer and how the observed aleatoric uncertainty (randomness) of complexity increases or decreases their uncertainty. When we discuss complexity, we mostly worry about how confusing its complicatedness will become, or is. A confusing complexity is a problem, and a clear or obvious complexity is a solved problem. Therefore, it is about "confusion", a state attributed to the predictive observer. Given these variables, a potential model could be formulated as follows:

$$Complicatedness\ \zeta \approx \frac{S^a {L_c}^{1/Cert}}{K^b} \quad (1)$$

- Complexity and its complicatedness (predictability) are relative to the awareness of the predictive observer. Also, the predictive observer's awareness influences a complex's complicatedness (predictability). The formula itself is generic, where factors are interpretable based on the ontology the user is using.
- *a and b* are parameters to be determined based on empirical data or expert judgement, which scale the influence of each factor on the complicatedness of an observed complexity to a predictive observer.
- Introducing an exponential dependency on Certainty indicates that even a tiny decrease in Certainty can significantly increase the likelihood of experiencing a confusing complicatedness. The value

of 1 can be given to a deterministic and linear complex (system); any value of less than 1 is for any system that cannot be certain 100% of (in other words, it possesses a grade of non-determinism).

- We use the operator "≈" instead of "=" to indicate that our metrics are holistic estimates that do not equate precisely to the mathematical expression. However, it can be indicated by a given mathematical expression. We believe the "≈" operator is more realistic than the "=" operator when it comes to dealing with systems modelling mathematics. It intends to model emergent systemic concepts that are not precisely equal to whatever the observer is viewing at the moment of observation. There is always something more that is not seen yet, so it includes the possibilities of some unknown related to the world, such that the mathematical model represents only a partial view. Thus, incorporating the sense of uncertainty of the observer into the formula, in other words, the observer's prior knowledge, also influences the observation, and it is being taken into consideration in the notation of "≈". By doing so, we indirectly acknowledge that "an observed whole, by a human observer, will always be more than what they assume to be the sum of observed parts". In systems, we are dealing with holistic concepts rather than an exact physical concept (like speed, weight, mass, etc).

3 Complicatedness of perceived formal model visual complexity

A Formal Model provides a language to describe systems concisely; it is, in fact, an information system about systems. The keyword in the last realisation is determining the building blocks of the "information" structure types in a formal model, which we will consider as the perceived complexity. Thus, we needed to answer the question: What are those recognisable parts and types of relations in every given formal model? In this case, we are considering Event-B formal models. There is at least one worldview we can look through at any formal model: the mathematical rigour behind its proof construct. However, this would have been a rather complicated approach. Since complexity is world-view-dependent, i.e., relative, it does not matter which way we look at the formal system; we can always analyse its complexity. The only distinctive factor between all possible views is the type of information directly associated with the intended utility. We can look at the number of letters, the number of dots or the amount of ink and paper that will cost us to print out the model on A4 pages with margins of 1cm around all edges, etc.; however, which method is meaningful and informative enough to indicate the level of impact on human observer cognitive stress?

To solve the dilemma of multi-choices in which all views can be correct, we looked at the nature of the interaction between a predictive observer (human) and the formal model. It can be noticed that the ability of the human observer to comprehend the model directly may be related to how complex and complicated it is to understand it. Upon observing multiple formal models, we notice that our first impression always starts with how they look visually (see Exhibit 1 and notice your first impression). Since observation is involved, "visual view" is an influencing attribute of the formal system. Hence, we considered visual complexity and its influence on model comprehension by a human observer to be the basis for model formal complexity. In other words, "if it looks too complex and complicated, then it is too complex and complicated".

One of the main reasons behind the introduction of bugs in any code is the misinterpretation of the purpose of the code. Unless the written code comprehensibly explains its purpose, those who will read it afterwards will have to spend time figuring out what the code is trying to achieve. One way to address it is through commenting, which can be viewed as a method to mitigate confusion and maybe could provide reviewers with some form of appreciation of how complex a source code is. As for formal modelling, relying on commenting alone to understand formal system components usually accompanies the implicit assumption that whatever the comment says, the formal code is the exact representation. We really need to see a simple formal code to quickly verify the comment context against it rather than rely solely on trust that the specialist has done the job correctly. Hence, commenting should also be considered when measuring the likelihood of confusion for a formal model. Considering the above reasoning, we can recognise that the following factors impact the likelihood of confusion due to the perceived visual complexity of a formal model (Event-B, to be specific):

- Variables.
- Hierarchy of refinements.
- Possible operation types represent available rules with which a system modeller can describe a system.
- Predicates in a refinement represent the rules describing variables' interactions.
- Comments associated with every step of formal code and any descriptive explanation.

4 Proposed Metrics Quantification Methods

The strategy we adopted in synthesising the metrics was mainly heuristic in nature; however, it was informed by established systems science concepts and mathematics philosophy. We underpinned the heuristics on the notion that formal methods (such as Event-B) intrinsically apply axiomatic systems, with invariants (theorems and axioms), events, and guards being viewed as a system of lemmas. The rationale behind such an assumption is that requirements are essentially predicates that can be evaluated in a binary fashion. Similarly, V. Ogleznev and V. Surovtsev (Ogleznev, 2018) draw an analogy that a government constitution is a form of an informal axiomatic system. The authors base this notion on the assumption that "Axioms are considered as contextual definitions of … concepts by means of which they are formulated". They argue that comparing a government constitution to an axiomatic system allows them to verify and validate its constructs against a well-founded criterion: Completeness, Consistency, and Independence. This is an interesting view on how formal modelling can contribute towards verifying and validating new laws before they may be introduced in the Constitution. Although the authors do not propose formalizing the constitution, it can be argued that laws have a systemic impact on the human population's livelihood. Since laws critically impact societal livelihood and well-being, it would make sense to consider formal verification and validation of laws before being constituted as part of the logical scrutiny process. From such thought experiments, we draw the analogy that system requirements are like a constitution to which the system must adhere. Since formal specifications describe those requirements mathematically, they, too, can be viewed as a rigorous representation of the system constitution.

In theoretical mathematics, axiomatic systems have a defined quality criterion. The term axiomatic system quality refers to how sound and well-founded an axiomatic system is. The quality of an axiomatic system is defined by three main properties: Independency, Completeness and Consistency (Partee, 1990). We attempted to quantify each criterion further and proposed the addition of a "likelihood of confusion of perceived complexity function". We derived the quantification from a self-defined requirement on what each quality metric equation should inform the evaluation. As for the quantification of the metrics, it is essential to know that we are not proposing a proving method for each criterion. Instead, we propose a method that quantifies how each criterion is achieved during formal modelling development based on the following risks to formal models' reliability and truthfulness (see Exhibit 2):

Exhibit 2. Mitigating risks related to formal modelling process by monitoring quality metrics performance

Risks to the reliability of the formal specification process	**Mitigation criterion**
Manual discharge of proof obligations (POs) may introduce human malicious or unintended errors (such as introducing bogus axioms like True ≈ False) only for discharging a PO.	We can measure the ratio of the number of POs automatically discharged to the total discharged POs. This measure can describe how Consistency and Completeness are achieved. Then, set a guideline for a good score based on average human cognitive capacity.

The reliability of a formal specification depends on the truthfulness of its axiomatisation of assumptions. Bogus axioms in a large, confusing formal specification are a real risk.	Measure the ratio of axioms to total invariants to expose the risk of introducing wrong axiomatisation of assumptions. The fewer axiomatisation of assumptions used, the more independent a formal specification is from axiomatised assumption. Then, set a guideline for a good score based on average human cognitive capacity.
Large, confusing formal specifications risk the reliability of quality checks by adding unnecessarily increased cognitive stress to the quality checker.	Measure the likelihood of confusing, complicated formal specifications based on factors related to best practice. Then, set a guideline for a good score based on average human cognitive capacity.

5 Evaluating the Performance of Formal Specification Reliability

We are utilising theoretical mathematics and being inspired by systems science to inform formal specification development of what a good model should look like. We are applying philosophy based on the notion that several expressions (operations, invariants, actions, etc.) compose an overall systemic visual complex that looks confusing or clear to a perceived observer. By including other factors such as completeness, consistency, and independence while caring about mitigating the likelihood of how confusing it can be, we can build a view of how model designers approached problems in hand and attempted to improve output quality, thus allowing us to evaluate the level of systems thinking involved.

We start by defining what Formal Model Reliability means. A formal model is said to be reliable when it is synthesised with no malicious corner-cutting approach (such as axiomatisation for the sake of discharging Proof Obligations POs) or when the specification is unnecessarily overcomplicated. Reliable formal modelling depends on three main factors: readability, complexity, and critical system thinking, in order to prove the correctness of specifications. We must have some properties to test performance against to quantify each factor. We derived four potential properties from Axiomatic Systems properties to help us specify each factor in the formal methods-based specification.

Firstly, we studied the independence of axioms; how did independent axioms impact model reliability by the formal system designer? We stipulate that the independence test can inform us of what kind of systems thinking the designer adopted to model the system and axiomatise assumptions. For instance, if every single invariant is modelled as a non-theorem (axiom) and more than 50% of them are redundant or derivable by other non-theorems, then the designer's mindset can be said to be focused only on discharging proof obligations at the expense of an increased number of non-theorem invariants. Formal specification reliability increases by decreasing the number of derivable non-theorems. It also increases by decreasing the total number of non-theorems in the model. This can be a helpful observation because it demonstrates that the designer could have decided to turn the 50% redundant axioms ratio into theorems to be proved. The designer then must justify why he or she made such a design decision. Without such a metric, it would be difficult to stipulate the number of redundant axioms across a large-scale model. The more redundant non-theorems are introduced, the poorer the proof of consistency claimed.

As for Consistency and Completeness, the main challenge we face is how to compare two models whose completeness and consistency would have been fully proved and verified. In other words, when all proof obligations are discharged, inherently, the model is proved to be 100% complete and 100% consistent. Since proof obligations (such as feasibility PO) test the consistency of each statement with the given axiomatic invariants and whether the action statements maintain the invariants, by the end of each model, the model will always be 100% complete and consistent. The problem arises: How can the models be compared regarding completeness and consistency if this is the case? The answer to this problem lies not in how complete or consistent the models are but rather in how completeness and consistency of the model

were achieved. We propose that any action, axiom, and theorem in the model should be completely provable automatically in the first instance by the tool used, for example, Rodin.

Therefore, we propose that the completeness and consistency test of formal modelling be based on the ratio of the number of POs discharged automatically to the total number of POs discharged interactively. We hypothesize that the designer's effort in discharging proof obligations is the quality measure of how reliable completeness and consistency were achieved. In other words, we define that when a designer manually discharges all proof obligations during the development phase, we run into the risk that some POs may have been maliciously discharged.

We also include the concept of complicatedness that can be influenced by how clearly the above criteria are performed and presented. However, this paper will consider the factors that influence the complicatedness of the perceived visual complexity as parameters for measuring the confusion associated with a developed formal model. We define the complexity of a model that specifies a simple system functionality in a confusing number of formal codes and utilises an unreasonable number of operational symbols across a minimal number of refinements as confusing. To mitigate cognitive confusion, we propose considering magic numbers related to the cognitive capacity of an average human to handle information (such as Miller's magic number 7). For example, the rule of anything above 3 (Doumont, 2002) would be considered confusing. In other words, a reasonable complicatedness of a given invariant, action, guard, etc., should not exceed 3 in one given expression. The rule of 3 is related to the cognitive comprehension of the human mind in handling conceptual complexity at a given time. We argue that overloading the human mind with excessive operations can lead to mistakes; for example, in Exhibit 1, a reviewer will likely be mentally unmotivated to go through a half-page formal specification. A reviewer may then base his / her review on trust that the associated comment is sufficient to explain what the formal code is intended to specify. This way, we run into the danger that the designer may incorrectly specify the functionality, and due to stress, he or she may have commented incorrectly.

5.1 Consistency and Completeness (1st Criterion)

We propose that the quality of achieving consistency and completeness (CC), combined together, can be evaluated by the ratio of the number of (proof obligations) POs that are proved automatically (AutoPOs) to the total number of POs in the model:

$$CC \approx \frac{\sum_1^n AutoPO}{\sum_1^n PO} \tag{2}$$

The metric informs that the more POs are automatically proved, the more reliably we can be assured that a model's consistency and completeness have been achieved. In other words, manual proof of POs risks human mistakes (intended or unintended); therefore, the manual proof method is not ideal.

5.2 Independency (2nd Criterion)

We propose that model Independency (Ψ) can be accounted for by considering non-theorems (axioms) in total rather than specifically for independent non-theorems only since reducing the number of all axioms, in general, will be compelling enough to ensure the reliability of the modelling process. We accept that axioms are independent as they ought to be. Since theorems are more desirable than non-theorems, as they need to be proved rather than be accepted as undeniable truths, we define the reliability of a model as a function of independence from the human-driven axiomatised assumptions by comparing the total number of theorems in a model to the number of total invariants (independent axioms & theorems).

$$\Psi \approx \frac{\sum_1^n theorems}{\sum_1^n invariants} \tag{3}$$

The metric should be interpreted as meaning that the more non-theorems introduced in the model, the less helpful the independence introduced for model reliability. The rationale behind this classification

is based on the observation that introducing an excessive number of non-theorems means a weaker system thinking Approach. We can assume that the designer aims to discharge POs at the expense of model reliability.

5.3 Complicatedness of formal model visual complexity (3rd Criterion)

Based on the theoretical foundation explained in sections 3 & 4, we reinterpret the generic function as follows: Given a formal model is a system, the complicatedness of a formal model is impacted by the following:

Scale of Perceived Formal Complexity (S): This reflects the number of Variables and the rules that define those variables and their relationships. In the case of Event-B-based formalism, it would be the sum of the number of variables (*Variables v*) in a given view and the number of formal expressions (*P* such as invariants, actions, guards etc) in the context and machines. It also includes the number of formal operations incorporated in the model (ω such as ∀,∈,⇒,∧ etc).

A refinement of a formal system is interpreted as a worldview of the formal model that raises the observer's awareness, thus increasing the predictability and understandability of the formal system (consequently, the clarity of the system being formalised too). The more refinements and tightly related those refinements are, the less confusing the complicatedness of the formal system (or the system being formalised). Therefore, the number of refinements (*Refinements* $\boldsymbol{\rho}$) is interpreted as the number of available views of a formal model to raise an observer's awareness of the system's complexity. Consequently, there is an inverse relationship between the number of refinements and confusion. We can capture this inverse dependency by using the power of the scale term. Given the latter analysis, we define:

$$S^{a} \approx (\nu * P * \omega)^{1/\rho} \tag{4}$$

Where: ν ≈ total number of variables, P ≈ total number of formal expressions, ω ≈ total number of formal operations symbols and ρ ≈ total number of formal views or refinements.

Knowledge about the Perceived Complexity (K): This factor accounts for what the observer already knows about Formal methods in general, the formal model itself and the number of words used in descriptive comments for each line of formal code, or associated documentation, in a formal model that traces the code to a meaningful aspect of the system with respect to the observer. The more description provided of what the code is doing, the less ambiguous it is for a code reviewer. Also, directly commenting on every line would further clarify the model. Therefore, we will include the word count of direct comments to each line as part of K. We assume all comments and associated descriptions are high quality in clarifying what each line means and how it directly models aspects of a requirement. The exponent b of the term K can have the following values (see Exhibit 3 and Equation 5):

Exhibit 3. Values of the b exponent of knowledge

Exponent b value	Observer's prior knowledge condition
0	**No prior knowledge**: If the observer has no prior experience or knowledge about formal methods in general, has no experience with Event-B language, and is not part of the presented formal model design team (no knowledge of the model itself).
1	**Some prior knowledge:** If the observer has some knowledge of formal methods and Event-B language but has never been part of the design team (so no prior knowledge of the observed formal model itself)
2	**Expert knowledge:** If the observer is an expert in formal methods, Event-B and knows the model as well as the original design team.

$$K^{b} \approx (\lambda)^{b} \tag{5}$$

Where: λ ≈ model overall explanation wordcount + direct commenting on each line of code wordcount, and b ≈ observer's prior knowledge about the domain and the model itself.

Latent Complicatedness (L_c): It can also be defined as latent confusion due to the complexity of the whole, where the observed complex view is part of it. In some systems, like engineered systems, such as language systems, grammatical rules are designed to cover all possible ways to covey or transmit meaning. However, practically speaking, making sentences that use all possible grammatical rules is rare. For example, in English, we do not use all rules in one sentence; we use only some. Just because we do not use all language rules in one sentence does not diminish the impact of the number of possible rules encompassed by the whole language on the likelihood that a yet-to-be-spoken sentence will be confusing. Some of the complexity of the whole set of rules in a language is still carried on by the perceived simple spoken or written sentence. This impact is captured by how likely the observer can be aware of or certain of all possible rules in the whole language, which is captured in the exponent of 1/Cert. For example, you can be a beginner in the Arabic language, and you understand one grammatical rule, but you are aware there are more rules that you are yet to be aware of. This awareness about the lack of knowledge of all Arabic grammatical rules still impacts the complicatedness of a spoken Arabic sentence to you. This is an implicit impact, and this is what we describe as Latent Complicatedness due to the awareness of potential aspects of an observed whole that the observer lacks knowledge about.

One may regard the reciprocal of certainty as, indeed, uncertainty. This is a legitimate consideration, which means, essentially, what we are taking into consideration when we measure complicatedness is the level of uncertainty associated with the size and nature of the whole, where any current observed view of its entire complexity is only part of. This is why we hear the concept of "the whole is bigger than the sum of its parts" It actually means "there is always a whole picture that is different from the currently perceived sum of parts". Essentially, it reflects the idea that (almost!) whatever view you have about something is only a partial view of what you may not yet be aware of. This is a natural intuition; after all, a situation is complicated when you feel uncertain of how true it is what you are observing. We can assume the following function:

$$Uncertainty \approx \frac{1}{Certainty} \tag{6}$$

Formula 6 can be read as "we can almost be certain that, in general, uncertainty is nearly equivalent to the reciprocal of certainty". After all, if we describe it as "=", then we are claiming that we are 100% certain of what general uncertainty in the universe is; however, by definition, the latter intuition is contradictory. For Event-B language, the number of rules would be the number of all possible operations the whole language offers to manipulate all possible relationships among variables, including the semantic methods that construct a formal specification. Those rules are somewhat certainly defined by the time any formal model is designed and do not change. Any observer of the Event-B formal model can be certain of the entire rule set of the language. In this case, the likelihood that Event-B rules as a whole may change is assumed to be 0. Therefore, the likelihood of certainty of the Event-B language system is attributed to 1.

So, what about Lc? Lc is the total number of all possible rules in the whole language that the observer is expected to be an expert in. When we count a rule in a formal language (for example, event-B), we count every singular atomic aspect, whether a mere definition of a notation or a type of relationship. Some aspects are methods to construct higher aspects; we count all the steps, including the higher aspects and general guidance rules. Based on the following reference (M´etayer & Voisin, 2009), we roughly estimate around 2000 distinct rules (based on a word document count of 2311 lines, assuming most lines contain at least a specific instruction, and some do not). This figure represents a property regarding the Event-B language as a whole and its impact on the observed formal complex view of concern. Therefore, we define the following latent confusion for the Event-B language system:

$$L_c^c \approx 2000 \tag{7}$$

Given the above, the adapted complicatedness function of any Event-B-based formal model ζ_B is:

$$\zeta_B \approx \frac{2000 * (\nu * P * \omega)^{1/\rho}}{(\lambda)^b} \tag{8}$$

We can define a metric for the readability of a single expression as a particular case of equation 1. However, when we consider the density of formalism in an expression, we average the total number of used operations and variables over the total number of expressions:

$$\overline{\zeta_{BExpression}} \approx \frac{\nu * \omega}{P * (\lambda)^b} \tag{9}$$

6 Formal Model Quality Performance Function

We interpret quality performance as how well the system designer handles trade-offs between complicatedness and expressiveness using systems thinking. The better system thinking (i.e. CC & $\Psi \approx 1$) and the lower the complicatedness is, the higher the formal specification model's effectiveness in supporting assurance. The final holistic Quality Performance for Event-B-driven models (Q_B) is:

$$Q_B \approx \frac{\mathrm{CC} * \Psi}{\zeta_B} \tag{10}$$

7 Cognitive Capacity Benchmark; the Rule of 1,2,3,9

Wright remarked on how much complexity a formal methods practitioner should introduce in every refinement step. He argued that it should be no more than a head full; however, quantifying "how much is a head full" is impossible (Wright, 2009). The latter argument may not be an absolute conclusion. It is essential to address this because comprehension errors due to program complexity (we argue "complicatedness") are recognised risks (Klemola, 2002). Since formal modelling's primary goal is to reduce error due to ambiguity, creating ambiguous and excessively complex formal specifications defeats its intended purpose. Therefore, by addressing the impact of information presented in models on human cognition, we can draw a more concise view of what good simplicity looks like. Miller started the agenda by linking cognitive psychology and information theory to propose how much information is manageable by a human mind (Baddeley, 1994). He proposed that the amount of data presented in each task manageable by human Short-Term Memory (STM) should be around 7 +/- 2 items. Throughout the years, the magic number 7 has been accepted by communication professionals as a good guideline. However, Doumont (Doumont, 2002) argues that 7 is in fact too many. The latter also labels, in terms of the human mind's capacity to handle information, 3 as the simplest complexity. We argue that what they are really examining is the "complicatedness" of presented informational complexity to a human cognitive capability. Furthermore, he argues that 1 is the focus, the symbol of consistency. Number 2 is symbolic, with the simplest form of redundancy. We argue that both opinions can be helpful in our definition of good formal modelling practices. Hence, we propose the rule of 1,2,3,9.

Considering the unavoidable impact of complexity on comprehension errors and what the above theorists suggested of what good enough can be, we propose the following schema of what good simplicity in formal methods looks like:

- Maintain ζ_B and $\overline{\zeta_{BExpression}}$ to be no more than 3**.**
- A given solution may have at least 9 levels of refinements (depending on the complexity of the task), but generally speaking, the formal specification should include more refinements or formal views of the system.
- For every refinement, ensure that no more than 1 new variable is introduced.

- For every refinement, ensure that no more than 2 new invariants are introduced.
- Start the abstract machine with 1 variable, 1 invariant, and 1 event.
- In every formal expression, ensure that no more than 3 operations are introduced.

The above guideline can be viewed as some instantiation of the rule of 1,2,3,9. We observe that to keep complicatedness down; we want to achieve less than a "head full". We can describe the general intuition to achieve a less than a head full using the complicatedness approximation formula ($less\ than\ a\ head\ full\ \zeta_B$):

$$\zeta_B \approx \frac{minmum\ language\ rules * (minimum\ v * minimm\ P * minimm\ \omega)^{\frac{1}{more\ refinements}}}{(more\ explaination)^{more\ knowledge\ about\ the\ model\ and\ Event-B\ language}} \tag{11}$$

8 Case Study

For the case study, we will use (M´ery, 2014) To validate our proposed criteria. The model utilised Event-B and Rodin tools to formalise several requirements related to aircraft landing gear systems. A total of 9 machines were developed, and roughly 11,236 operations (the total count of the following operations types: (:≈, ∈, ≈, ⇒, ∧, >, →, ↦,→,x-1, |,∀, (), {},×,..,[]) and roughly 47 unique variables. The model contains roughly 92 invariants, 1397 actions, and 1377 guards; thus, the total of applicable formal expressions sum up to 2866 predicates. The developers do not specify how many invariants are set to non-theorems, thus invoking a valuable question as we cannot evaluate the second criterion without such information; in this case, we will assume $\Psi \approx 1$. Moreover, developers specify 529 proof obligations, 448 of which are discharged automatically. We are not counting the model contexts in this example due to their absence from the document; typically, contexts are included in the evaluation. In their full report, they described the whole model in 9 machines. We counted the number of words in natural language (excluding titles) that directly explain each machine. The team did not describe the contexts they used to support each machine. The word count came up to 2619 words to explain the model. The design team did not show any direct comments on each line of formal code, thus it is assumed to be 0.

8.1 Consistency and Completeness (1st Criterion)

We can evaluate how CC is achieved by applying equation (2):

$$CC \approx \frac{448}{529} \approx 0.85$$

The figure above informs the development team that the model's Consistency and Completeness are 85% reliable. One reason could be the excessive, confusing complexity introduced in the model. Therefore, a simpler model could achieve higher CC reliability.

8.2 Formal Model Complicatedness (3rd Criterion)

By applying equation (8), we assign b a value of 1, assuming the reviewers of the model are knowledgeable about formal methods and Event-B. However, they are not familiar with the model itself. Since we are knowledgeable in formal, we stipulate:

$$\zeta_B \approx \frac{2000 * (47+2866+11236)^{1/9}}{2619} \approx 8$$

The average readability (density of formalism) of every line of formal code can be determined using equation 9. We must estimate how much natural language explanation was spent per line of code. Since the design time did not include lines of comments to every line, we will rely on devising the number of words used to explain the model by the number of expressions $\lambda \approx (2619/2866) \approx 0.75$. We assumed $b \approx 1$:

$$\overline{\zeta_{BExpression}} \approx \frac{47*11236}{2866*0.75} \approx 245$$

The overall quality of the model is determined using equation 10:

$$Q_B \approx \frac{0.85*1}{8} \approx 0.1$$

References

Ardito, L., & al., e. (2020). A Tool-Based Perspective on Software Code Maintainability Metrics: A Systematic Literature Review. Scientific Programming.

Ashby, W. Ross, "Some peculiarities of Complex Systems", Cybernetic Medicine, Vol 9, no. 1, (1973).

Baddeley, A., "The magical number seven: Still magic after all these years?," Psychological Review, vol. 101, no. 2, pp. 353–356 (1994), doi: 10.1037/0033-295x.101.2.353.

Barsamian, M. Introduction to Axiomatic Geometry. Athens Ohio, USA: Ohio University, (2017).

Corry, L., David Hilbert and the axiomatization of physics (1898-1918) : from Grundlagen der Geometrie to Grundlagen der Physik. Dordrecht: Kluwer (2004).

Douce, C., Durant, S. Understanding program complexity: an approach for study. Psychology of Programming Interest Group Annual Conference (2011).

Doumont, J. L. "Magical Numbers: The Seven-Plus-or-Minus-Two Myth," IEEE Transactions on Professional Communication (2002).

Halstead, M.H., Elements of software science, Elsevier, New York (1977).

Hummel, B. McCabe's Cyclomatic Complexity and Why We Don't Use It. Garchingbei München: CQSE GmbH (2014).

Hao, G., & al., e. (2023). On the accuracy of code complexity metrics: A neuroscience-based guideline for improvement. Frontiers in Neuriscience. doi:https://doi.org/10.3389/fnins.2022.1065366

Kearne, J. K. (1986). Sfotware Complexity Measurement. Communications of the ACM, 1044-1050.

Klemola, T., Rilling, J., "Modeling Comprehension Processes in Software Development," Proceedings First IEEE International Conference on Cognitive Informatics, Calgary, AB, Canada (2002).

Klir, George J. Facets of Systems Science, Vol 15, 2nd ed, State University of New York, IFSR international series on systems science and engineering, Springer Science + Business Media, LLC. (2001).

Lee, C. W. Axiomatic Systems, Kentucky, USA: University of Kentucky (1997).

M´ery, D., Singh, N. K., "Modeling an Aircraft Landing System in Event-B," in ABZ 2014: The Landing Gear Case Study, pp. 54–159 (2014).

Miller, G. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information, Vol. 101, No. 2, 343-352. The Psychological Review (1956).

Ogleznev, V., Surovtsev, V. "The Constitution as an Axiomatic System," in Axiomathes, Aug. 2017, vol. August, no. 28, p. pages 19–232 (2018).

Partee, B. H., Wall, R. E., Meulen, Alice Ter, Mathematical methods in linguistics. Dordrecht Etc.: Kluwer Academic (1990).

Romanovsky, A. & Thomas, M. Industrial Deployment of System Engineering Methods, Springer (2013).

Snook, C., Butler, M. UML-B: Formal modelling and design aided by UML, ACM Transactions on Software Engineering and Methodology (2006).

Tashtoush, Y., & al., e. (2014). The Correlation among Software Complexity Metrics with Case Study. arXiv.

Watson, A. H. & Mccabe, T. J. Structured Testing: A Testing Methodology, Gaithersburg: National Institute of Standards and Technology (1996).

Wright, S. Formal Construction of Instruction Set Architectures, University of Bristol, Bristol, UK (2009).