

# MAV-Vis: A Notation for Model Uncertainty

Michalis Famelis and Stephanie Santosa  
University of Toronto, Canada

**Abstract**—We apply the “Physics of Notations” theory to design MAV-Vis, a concrete syntax for partial models, i.e., models where design uncertainty is explicitly captured. To validate our implementation of this theory in creating MAV-Vis, we designed and executed an empirical user study comparing the cognitive effectiveness of MAV-Vis with the existing, ad-hoc notation, MAV-Text. We measured the ease, speed, and accuracy of each notation for reading and writing partial models.

## I. INTRODUCTION

Software modelers must often work with design uncertainty, i.e., incomplete information about the content of models. This can be challenging, as they must comprehend the nature of uncertainty and work around multiple design possibilities. We have introduced *partial models* as a way to explicate and handle design uncertainty [14]. A partial model consists of a “base” model, augmented with annotations to express a set of possible conventional models. We can thus capture the uncertainty that a modeler has about choosing one of these possibilities. The base model can conform to any metamodel; so partial modeling is language-independent.

Using partial models, we have shown how to model uncertainty, reason in its presence, systematically remove it using refinement, propagate it, manipulate and transform models that contain it, etc. [14], [4], [3]. This provides ample evidence that partial models allow us to efficiently do *automated* reasoning in the presence of uncertainty. However, apart from being formal, machine-processable artifacts, partial models are (like all models) also a means for human communication. Thus partial models should be an effective means for *expressing* uncertainty to others and for *understanding* the uncertainty expressed by others. Both of these communication tasks are greatly dependent on *notation* [8].

While partly influenced by work on behavioral modeling [7], the text-based notation (called MAV-Text) used in existing work has been developed in an ad-hoc way, without much attention to usability and communication effectiveness. As working with uncertainty poses many challenges in itself, it is important that the notation used to express it does not present additional barriers in communication.

In this paper, we develop a notation for partial models focusing on usability, which is defined as *cognitive effectiveness*, i.e., ease, speed and accuracy of use [6]. We utilize D. Moody’s “Physics of Notations” theory [8], which provides principles for designing cognitively effective notations, to analyze MAV-Text, and form a basis of our design rationale for a new syntax, called MAV-Vis. To validate our implementation of Moody’s principles in designing MAV-Vis, we performed a user study, where we empirically evaluated the cognitive effectiveness of MAV-Vis against MAV-Text.

**Related Work.** Moody [8] provides a survey of relevant background work to visual notation theory and introduces the

principles for visual notation design that we apply in our syntax assessments. This has been applied to assess the *i\** visual notation [9] and to evaluate the syntax of UML [10]. There has also been prior work in empirical syntax evaluation [1]. A notable example is the design of an experiment to compare between two notational alternatives for process modeling [5].

In Sec. II, we introduce partial modeling using an example, and present an analysis of MAV-Text. We present MAV-Vis in Sec. III. In Sec. IV we describe the empirical validation of our design with a user study and conclude in Sec. V.

## II. MOTIVATING EXAMPLE

Partial models are formalized using *MAVO partiality* [14] which supports four kinds of annotations: (a) *May partiality*: annotating a model element with *M* indicates that we are unsure whether it should exist in the model or not. Additional dependencies between groups of may elements can be expressed in a propositional *May formula*. (b) *Abs partiality*: annotating an element with *S* indicates that we are unsure whether it should actually be a collection of elements. (c) *Var partiality*: annotating an element with *V* indicates that we are unsure whether it should actually be merged with other elements. (d) *OW partiality*: annotating the entire model with *INC* indicates that we are unsure it is complete. Within this framework, we refer to a specific decision about which the modeler is uncertain as a *Point of Uncertainty* (PoU).

We show an example of a partial Entity-Relational [2] model in Fig. 1. The model describes a hotel management system where Customers reserve Rooms, while explicit Access must be granted for entering a Room. The model has four PoUs:

PoU1: The modeler is unsure whether Customer and Employee should have a common superclass Person from which to inherit the attributes name and surname.

PoU2: The modeler is unsure about associating Access with Customer in particular or Person in general.

PoU3: The modeler is undecided about the *securityAttributes* of the entity Access.

PoU4: The modeler is unsure which entity should own the property *internetAccess* and what the *id* attribute of such an entity-with-internet should be.

In our example, the modeler is sure about the completeness of the model (no *OW* partiality). The points of uncertainty PoU1 and PoU2 are explicated in Region II of the model, using *May* partiality. They are accompanied by a *May formula*, shown in the bottom of Region II, which specifies their dependency. The points of uncertainty PoU3 and PoU4 are explicated in Region I, using *Abs* and *Var* partiality respectively.

The partial model in Fig. 1 is expressed in MAV-Text: the notation introduced in [14], incorporating from [4] the notation for expressing *May* formulas. In MAV-Text, MAVO annotations are expressed textually, and dependencies between

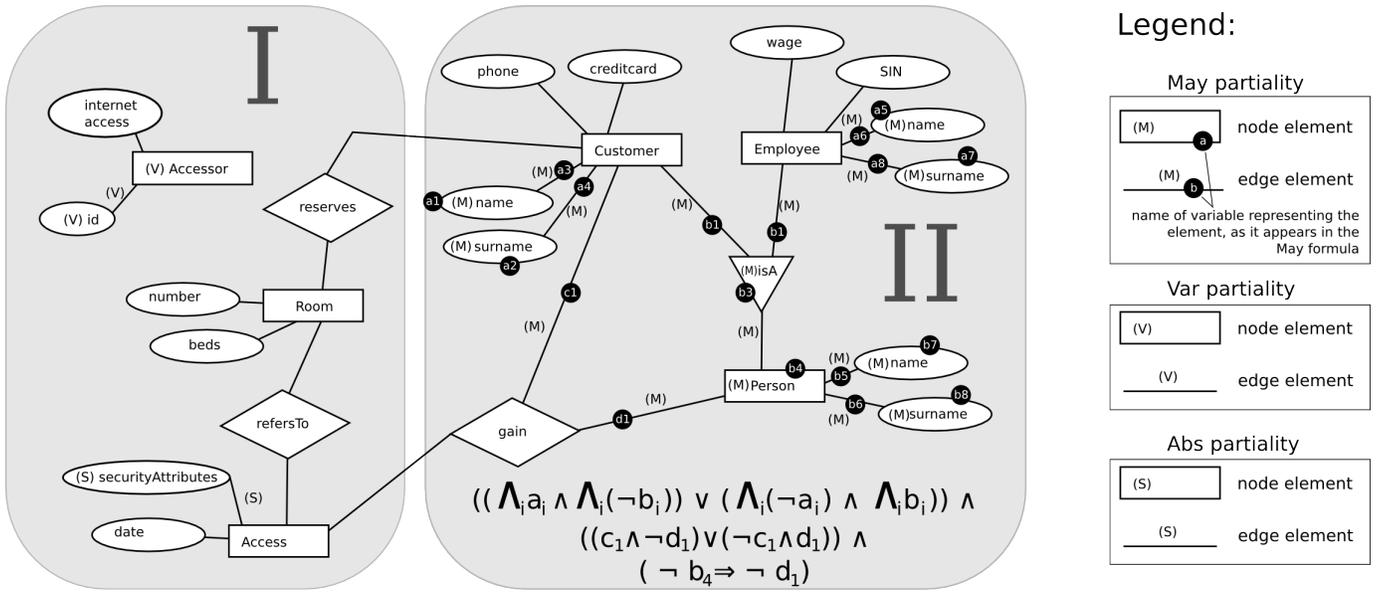


Fig. 1. Motivating example in MAV-Text syntax. Region I: elements with Abs and Var uncertainty. Region II: elements May uncertainty and May formula.

TABLE I  
ASSESSMENT OF SYNTAXES MAV-TEXT AND MAV-VIS.

Design principle [8]	MAV-Text Assessment Summary	MAV-Vis Assessment Summary
<b>Semiotic Clarity</b> (1:1 symbol-concept correspondence)	++ Each annotation has one-to-one correspondence to meaning. Note: different text annotations considered distinct here, despite being homographs.	++ One-to-one correspondence to meaning.
<b>Perceptual Discriminability</b> (Distinguishable symbols)	-- All notations, other than circles on uncertainty element labels, are text-based: zero visual distance between them.	++ Different retinal variables for each notation: shape and texture helps differentiate between uncertainty types (dashes, icons, piles).
<b>Semantic Transparency</b> (Representation suggests meaning)	- Annotations not easily associated with concepts. Relationships across may elements not visible; relies only on May formula.	+ Representations reflective of concepts. Relationships are visible.
<b>Complexity Management</b> (Explicit mechanisms handling complexity)	- New annotation element introduced for each element with uncertainty. No mechanism for chunking information to manage additional complexity.	++ Grouping applies uncertainty to entire submodels (not per element).
<b>Cognitive Integration</b> (Global view from different diagrams)	No specific mechanisms, but the May formula contextualizes May elements to overall uncertainty.	No specific mechanisms, but the May groupings and dot notation contextualize may elements to the overall uncertainty.
<b>Visual Expressiveness</b> (Symbols denoted with multiple visual properties)	-- All textual encoding: measures to zero-degrees of visual expressiveness.	++ Visual expressiveness = 4. <i>Shape</i> : icons and piles. <i>Color</i> for PoU's. <i>Texture</i> : dashed line treatment. <i>Size</i> : may dependencies.
<b>Dual Coding</b> (Text complements graphics)	-- No dual coding; May formula is separated with no visual connection with referent elements: principle spatial contiguity violated.	++ Color and text used together. In-place annotations for spatial contiguity.
<b>Graphic Economy</b> (Use of manageable number of symbols)	++ Not an issue - no use of graphic symbols.	+ High visual expressiveness keeps cognitive manageable (never exceeds 6 symbols per visual variable).
<b>Cognitive Fit</b> (Suitable for audience and medium)	+/- Appropriate for representational mediums of pen-and-paper and tools. Requires skill in propositional logic for May formula.	+ No specialized skills required. Pen-and-paper appropriate.

PoUs are indicated in the May formula using propositional variables, shown in the model using letters in black circles.

MAV-Text may not be the most appropriate notation for maximizing the reader's cognitive effectiveness. This assessment can be theoretically grounded using the design theory in [8], which outlines a set of principles for designing effective visual notations. The results of our assessment are summarized in Table I where we identify several issues with MAV-Text. For each principle, we assign MAV-Text a rating, ranging from ++ ("very good") to -- ("very bad"). We additionally use +/- to mean "it depends". No rating means that the notation

is neither good nor bad. MAV-Text has very good Semiotic Clarity and Graphic Economy and may have good Cognitive Fit depending on the context. However, it ranks poorly on almost all of the other principles, most notably Perceptual Discriminability, Visual Expressiveness and Dual Coding.

### III. DESIGN OF MAV-VIS

Our analysis of MAV-Text highlights issues to be addressed in the development of our new notation, called MAV-Vis. Our goal with MAV-Vis is to create a cognitively effective syntax for uncertainty, and we apply the principles in [8] to methodi-

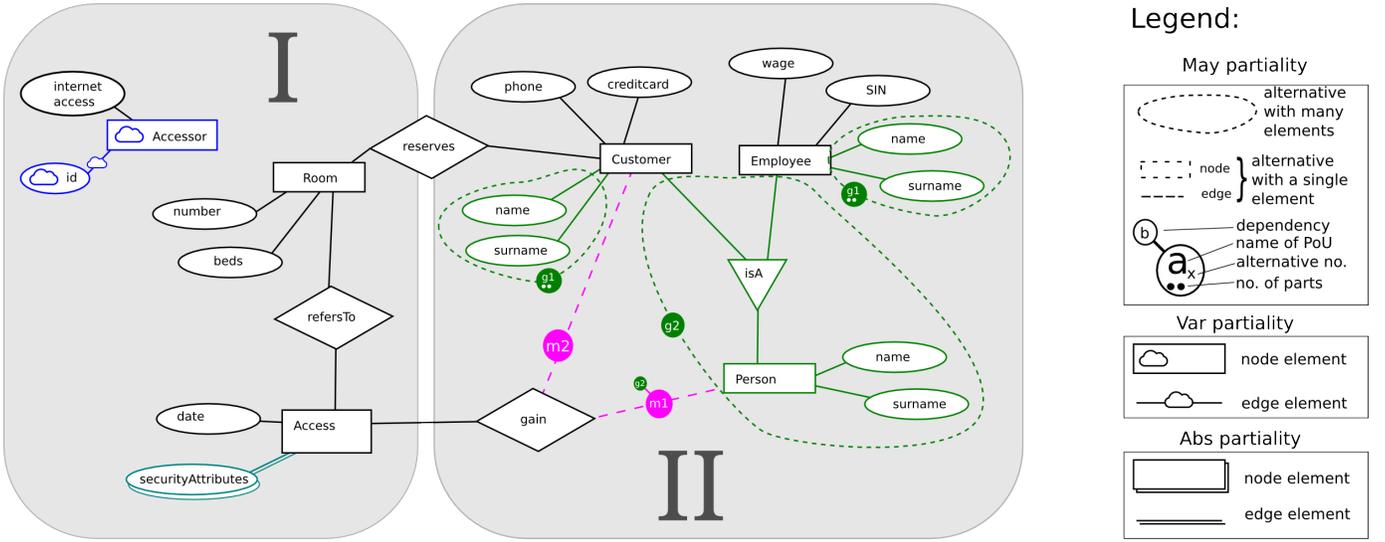


Fig. 2. The model from Fig. 1, expressed in MAV-Vis. Each PoU is given a unique color.

cally guide the design and assess the resulting notation. Here, we briefly describe the resulting MAV-Vis syntax for partial models and our design rationale behind it. For comparison, the example from Fig. 1 is shown using MAV-Vis in Fig. 2. We summarize our heuristic assessment of MAV-Vis with respect to the theory of “Physics of Notations” in Table I.

**Points of Uncertainty** are distinguished by color. In particular, each design decision about which the modeler is uncertain is given a unique color. In Fig. 2, the elements associated with PoU1 are colored green, those associated with PoU2 are colored magenta, those associated with PoU3 are cyan, and those associated with PoU4 are blue. Since the notion of a PoU serves mainly to chunk information for easing comprehension, losing color information (e.g., in a printout) does not remove essential information from the model.

**Var Uncertainty** is represented by a *cloud* icon, as shown in Region I of Fig. 2. The idea here is to use shape for visual discriminability and to select a representation that can be associated with the concept, in the form of a sketchable icon: a pen-and-paper friendly symbol that requires minimal drawing skills so that the Cognitive Fit is appropriate.

**Abs Uncertainty** is represented using a *pile* metaphor, as shown at the in Region I in Fig. 2. Piles are an intuitive visual notation to denote a set, improving Semantic Transparency, Visual Expressiveness, and Perceptual Discriminability.

**May Uncertainty.** For each PoU, each allowable configuration of the May-annotated elements corresponds to an *alternative*, i.e., a distinct way to concretize the partial model. In MAV-Vis, alternatives are expressed as first-class entities. An alternative  $A$  is indicated by enclosing its elements with a dashed line. If  $A$  only has one element, its border can be dashed instead. Each enclosure is given a label of the form  $x_n$ , where  $x$  is the name of the PoU and  $n$  is the ordinal number of  $A$ . If  $A$  is not spatially contiguous, the label is also accompanied by a number of dots representing the total number of clusters, to help readers quickly recognize non-contiguous alternatives. In Fig. 2, PoU1 is represented by the color green and has two alternatives,  $g_1, g_2$ . The alternative  $g_1$  has two separate

clusters, so its labels have two white dots. Dashed lines and enclosures improve Visual Expressiveness, adding the Texture visual variable to the notation and making the elements more easily distinguishable. Additionally, the in-place grouping and identification scheme of May elements exhibits Dual Coding with color and prefix reflecting the PoU, and there is spatial contiguity between the May elements and their combinations. Formally, each element in a dashed line is annotated with  $M$ . Assuming that a partial model  $M$  has  $k$  PoUs  $\{P_1, \dots, P_k\}$ , that a given PoU  $P_x$  has  $n$  alternatives  $\{A_1^{P_x}, \dots, A_n^{P_x}\}$  and that a given alternative  $A_y$  has  $l$  model elements, explicating May uncertainty as described above means that the May formula  $\phi_M$  of  $M$  is:  $\phi_M = \bigwedge_{x=1}^k \phi_{P_x}$ , where  $\phi_{P_x} = \bigoplus_{y=1}^n \phi_{A_y}$  where  $\phi_{A_y} = \bigwedge_{z=1}^l e_z, e_z \in M \wedge e_z \in A_y$  and  $\bigoplus$  denotes propositional exclusive-or.

**Dependencies between Alternatives.** Partial models containing May uncertainty can also include constraints in propositional logic [4], expressing allowable configurations of May elements. Creating a visual language for full propositional logic would be difficult, if not self-defeating. Instead, we opted for a syntax that allows users to graphically express simple dependencies similar to the “requires” dependencies of feature models [16]. The syntax can be extended to represent more complex dependencies.

In Fig. 2, the modeler cannot choose the *dependee* alternative  $m_1$  of PoU2 (shown in magenta) without choosing the *dependum* alternative  $g_2$  of PoU1 (in green). This is indicated by adding to the label of the dependee a link to a small version of the label of the dependum. This notation allows the modeler to express dependencies locally and intuitively. In a partial model  $M$ , a dependee alternative  $a_d$  may have a set of dependum alternatives  $\{d_1, \dots, d_N\}$ . Explicating these relationships as described above means that for each dependee  $a_d$  the May formula  $\phi_M$  is enhanced by conjuncting the expression:  $\bigwedge_{x=1}^N (\neg d_x \Rightarrow \neg a_d)$

**Limitations.** MAVO is a set of formal annotations that can be used with arbitrary modeling languages. It is thus impossible to guarantee Semiotic Clarity (1:1 correspondence between

symbols and concepts), since there can be no guarantee that the base language does not use the same graphical constructs to denote its own concepts. We have used MAV-Vis with UML Class Diagrams [13] (not shown in the examples in this paper) and Entity-Relationship (E-R) Diagrams [2]. It is easy to adapt the notation to other languages but the process cannot be easily automated. However, in MOF [12], Class Diagrams are used to express any model in its abstract syntax. Thus, even though MAV-Vis may not be able to annotate arbitrary concrete syntaxes, it can always be used with abstract syntax.

Finally, MAV-Vis does not address the OW partiality. OW is expressed at the model level and so the annotation language would have to be either explicitly combined with tooling or expressed using formalisms that support multiple abstraction layers, like macromodeling [15].

#### IV. EVALUATION

As shown in Table I, MAV-Vis significantly improves on the issues identified with MAV-Text. Our methodological approach guarantees that all design principles were considered in the creation of MAV-Vis. To evaluate our implementation of the theory, we performed usability testing with real users — an essential component of usability practice [11]. We followed the empirical research techniques outlined in [1] that adapt usability testing to software modeling languages. We conducted an experiment to evaluate and compare the cognitive effectiveness of MAV-Text and MAV-Vis, as determined by three measures: *ease*, *speed*, and *accuracy*. Our goal was to confirm or refute the hypothesis that MAV-Vis improves the score for each of these measures compared to MAV-Text.

##### A. Setup

**Design.** We designed a series of software modeling tasks with the goal of measuring the cognitive effectiveness of each syntax. Each participant was asked to start with the Free-Form task of writing uncertainty into a model using ad hoc notations, and was given the liberty to invent them as needed. This task served as a “warm up” to the uncertainty concepts, giving participants opportunity to ask questions about MAVO. This task also provided insight into the types of notations people would naturally use to communicate these concepts. Each participant was then given a reading and writing task using

one syntax type, followed by a reading and writing task using the other syntax type. Each task involved all 3 uncertainty types addressed by the notations (Abs, Var, and May plus constraints), in multiple points of uncertainty (PoU’s). Table II describes each modeling task.

Task A was performed on a simple E-R diagram modeling a blog, while Tasks B and C were based upon richer modeling scenarios. The same base model was used for reading and writing within a task. Two modeling scenarios were used (one for each syntax) to support the reading and writing tasks: School Personnel (UML Class Diagram) and Hotel Administration (E-R Diagram, shown in Fig. 1).

We used a within-subjects design, to reduce selection bias and allow for each participant to compare notations and express their preferences. We controlled for two independent variables: the order in which the syntaxes are presented and the model scenarios used for each of the syntaxes. These were counter-balanced in a 2x2 Latin square.

We measured cognitive effectiveness with respect to speed, ease and accuracy [8], determined by task completion time, questionnaire responses, and error counts and comprehension scores, respectively. For the effect of notation on accuracy we tried to observe (a) syntax correctness, and (b) the effect on comprehension and correct communication of uncertainty.

**Procedure.** Participants were given a background questionnaire to collect information on their areas of expertise and prior knowledge of MAVO uncertainty, and experience levels with UML and E-R diagrams. They were then given an 8-minute tutorial on May, Abs, and Var uncertainty concepts, including definitions for the terms “Point of Uncertainty” and “concretization”. Participants all started with the Free Form notation (Task A). Following this, they were given a summary sheet explaining the first syntax type to read and use as a reference for the tasks. Using that syntax, they completed a reading and a writing task for the modeling scenarios described earlier (B1-B2 or C1-C2), repeated with the other syntax type and scenario. At the end, participants filled out a post-study questionnaire rating both notations for each uncertainty type, and indicating which syntax was preferred and why. At the start and end of all tasks in Table II, participants recorded the displayed time.

TABLE II  
MODELING TASKS OF THE EVALUATION.

Task	Name	Description	Uncertainty
A	Free-form (Write)	Simple base model given. Participant asked to add indicated PoUs	3 PoUs: One May PoU with 2 alternatives. Then one Abs PoU and one Var PoU, both conditional on one of the alternatives in the May PoU.
B1	MAV-Text (Read)	A model is presented with uncertainty given in MAV-Text. Participant asked to circle PoUs and indicate what the designer is uncertain about. Then asked to draw one example concretization for the Abs and Var PoUs and all concretizations for the May PoUs.	Four PoUs: one Abs, one Var, and two May with layered dependency (the second PoU is conditional on one of the alternatives in the first.)
B2	MAV-Text (Write)	Model from B1 given with uncertainty resolved. Participant is given a written scenario describing uncertainty, and asked to use MAV-Text to express the uncertainty in the model.	Three PoUs are given: one Abs, one Var and one May with 2 alternatives.
C1	MAV-Vis (Read)	A model is presented with uncertainty given in MAV-Vis. Participant asked to circle PoUs and indicate what the designer is uncertain about. Then asked to draw one example concretization for the Abs and Var PoUs, and all concretizations for the May PoUs.	Four PoUs: one Abs, one Var, and two May with layered dependency (the second PoU is conditional on one of the alternatives in the first).
C2	MAV-Vis (Write)	Model from C1 given with uncertainty resolved. Participant is given a written scenario describing uncertainty and asked to use MAV-Vis to express the uncertainty in the model	Three PoUs are given: one Abs, one Var, and one May with 2 alternatives.

TABLE III  
EASE RESULTS.

		Intuitive	Easy to Remember	Efficient to Read	Efficient to Write	Number Preferred
Abs	MAV-Text	3.2	3.3	3.3	3.6	1
	MAV-Vis	<b>4.2</b>	<b>4.2</b>	<b>4.3</b>	3.6	<b>10</b>
Var	MAV-Text	2.5	2.8	3.3	<b>3.5</b>	2
	MAV-Vis	<b>3.3</b>	<b>3.7</b>	<b>3.8</b>	3.2	<b>8</b>
May	MAV-Text	2.9	3.6	3.2	3.2	4
	MAV-Vis	<b>3.8</b>	<b>4.1</b>	<b>3.9</b>	<b>3.8</b>	<b>7</b>
May Groupings	MAV-Text	2.8	3.2	2.6	3.3	3
	MAV-Vis	<b>3.7</b>	<b>3.8</b>	<b>3.5</b>	3.3	<b>8</b>

TABLE IV  
SPEED RESULTS.

Speed	Reading (mm:ss)	Writing (mm:ss)
MAV-Text	14:06	9:29
MAV-Vis	<b>11:58</b>	9:42

TABLE V  
ACCURACY RESULTS.

(a) Read	Abs (score/6)	Var (score/6)	May (score/6)	Total (score/6)
MAV-Text	3.9	5.1	2.8	11.8
MAV-Vis	3.9	5.2	<b>4.2</b>	<b>13.3</b>

(b) Write	Syntax (error count)	Comprehension (error count)
MAV-Text	<b>2.3</b>	1.7
MAV-Vis	3.0	1.7

**Participants.** Twelve unpaid participants took part in the study. All participants had a Bachelors degree or higher in Computer Science and nine were specialized in software engineering. The average experience level with MAVO uncertainty was 2.2 out of a 5-point Likert scale; however, three were experts in MAVO uncertainty and were already comfortable with MAV-Text. The average experience level for UML diagrams was 3.3, and for E-R diagrams 2.9.

**Apparatus.** The experiment was performed with pen and paper. Each participant was given four pens (black, green, blue, and red) with an experiment packet containing colored printouts and asked to proceed through it in order. A clock showing the time in seconds was displayed on a projector for the participants to use as reference for recording task times. Our apparatus and collected data are available online at <http://www.cs.toronto.edu/famelis/datasets.html>.

## B. Results

Due to the limited number of participants, there was not enough data to perform statistical analysis. We report here general observations on the quantitative measures as well as qualitative feedback from participants. The observations for ease, speed and accuracy are shown in Tables III, IV, and V, respectively. Figures in bold indicate observations that were clearly in favor of one of the two notations.

**Ease.** For Abs, Var, and May uncertainty, as well as May groupings, Participants were asked to rate how intuitive, easy to remember, efficient for reading, and efficient for writing the corresponding representations of each syntax type were. We used a 5-point Likert scale, with 1 representing strong disagreement and 5 representing strong agreement. Thus, higher numbers indicate more successful results. Participants

were also asked to indicate which syntax they preferred for expressing each uncertainty type, as well as which syntax they preferred overall. The results are summarized in Table III.

Participants favoring MAV-Vis mostly found it clear and easy to associate with semantic meaning. One participant commented that it “conveys much more info and [is] easy to disambiguate [symbols]”. Another indicated that MAV-Text “annotations are very similar so meanings are almost overloaded. I had to think to remember the meanings”.

**Abs:** Several participants preferring MAV-Vis commented that it was “quick”, “easier to notice”, and “intuitive”, making comments such as “I could get it at the first glance” and that it “clearly denoted that it was a collection”.

**Var:** Participants were divided on the appropriateness of the cloud icon. Some expressed it was easily associated with the Var concept, while others did not find it intuitive, stating for example “cloud does not equal var in my head”. Two of the participants indicating this still stated a preference for MAV-Vis due to its visual appeal and ability to “stand out more”. This issue with the semantic association of the cloud symbol was the reason one participant preferred MAV-Text. The other participant preferring it stated that it takes longer to draw the cloud and “takes up too much space”.

**May:** Many participants found dashed lines to have semantic clarity, and indicated that it was easier to see and read. Some participants however, indicated that they found the (M) annotation to be cleaner, and easier to draw.

**May Groupings:** Participants liked that MAV-Vis provided a way of “grouping and visualizing all the choices simultaneously”. Most participants were familiar with propositional logic. One participant noted that as a result, “the learning curve [for MAV-Vis] is a bit steep, but it makes sense and [is] way easier than a formula”. Others commented that the May formula was “more powerful” and “commonly known”.

**Speed.** Table IV summarizes the average completion times for each task. As expected, reading tasks took less time to complete using MAV-Vis, with MAV-Text averaging 2:08 min longer to complete (17.8% longer completion time). There was also a substantial amount of overhead in these tasks, since users were required to demonstrate their comprehension by drawing concretizations and writing out descriptions of the uncertainty. As this overhead was consistent across syntax types, we can attribute the time difference to the portion of task time used for processing the information in the diagrams. This suggests that the impact on comprehension speed is actually much greater than the time difference measured.

**Accuracy.** Reading performance for each type of uncertainty was evaluated using the success rates for correct identification of PoUs, correct identification of the underlying uncertainty, and correct drawing of concretizations. Each of these was given 2 points if correct, 1 point if partially correct, and 0 otherwise. Table V(a) summarizes average score results. MAV-Vis scored much higher for May uncertainty and grouping, with an average score of 4.2 (versus 2.8 in MAV-Text).

For writing we counted the total number of syntax and conceptual comprehension errors. Table V(b) shows average error counts in both categories for each syntax. The number of syntax errors for MAV-Vis (3.0) was greater than those for MAV-Text (2.3). However, 1.7 errors in MAV-Vis were attributable to incorrect/absent use of color-coding for grouping uncertainties, which is not a critical error.

**Threats to Validity.** Several factors could have impacted the results. Our sample size (12 participants) did not provide us with enough data to perform statistical analysis. Nonetheless, it provides strong indicators on how the syntaxes compare. Moreover, 3 participants reported extensive prior exposure to MAVO, which could have introduced some bias towards MAV-Text. Also, an imbalanced knowledge of UML vs E-R can result in some selection bias with our small sample size.

We also note that there were varying levels of proficiency with propositional logic amongst our participants. Since all were computer scientists, this variation is likely to be reflective of the target audience for the syntax. Additionally, some participants may have had difficulty with the underlying uncertainty concepts. This may have had an impact on accuracy metrics; however, it would have had an equal effect on both syntaxes.

### C. Discussion

Piles were intuitive representations for expressing Abs partiality and improved writing ease. No difference was measured in accuracy but this greater semantic transparency likely contributed to the improved reading speed. We noted that one participant in the Free-hand task independently came up with the same notation. Opinions on using the cloud icon to represent Var partiality were polarized. However, people who complained about semantic disconnect still preferred it to the (V) annotation, owing to its visual pop out. We believe that simple icons are an appropriate solution for indicating uncertainty modifiers, even if the particular choice of icon is disputable. Dashed lines were intuitive for May partiality. There is, however, a trade-off between using propositional logic (a language that many computer scientists are already familiar with) and introducing something new requiring some learning in exchange for the added visualization power.

Overall, the results favor MAV-Vis, indicating that graphical elements improve cognitive effectiveness (speed, ease, accuracy), especially for reading. For writing, the notations rated similarly, slightly favoring MAV-Text in accuracy. Color-coding PoU errors were the main differentiator of accuracy, so we can consider these results to be very similar as well since PoU is not a formal MAVO concept and is not semantically critical for Var and Abs. Also, since May alternatives are dual-coded, diagrams can be unambiguously read without color, using the prefix labels.

While the clear majority of participants preferred MAV-Vis, this may not indicate that it is a universally better solution.

Different learning style and expertise may yield different preferences. As the Cognitive Fit principle suggests, a “one-size fits all” solution is not often ideal, and this accounts for some of the variation in our results.

## V. CONCLUSION

We studied the notations for expressing design uncertainty using the design theory in [8]. We assessed the existing, ad-hoc notation for partial models, called MAV-Text, and designed a new notation, called MAV-Vis. To assess the validity of our implementation of the theory, we conducted a user study to collect empirical data on the cognitive effectiveness (speed, ease and accuracy) of the two notations. We found that, overall, users preferred MAV-Vis, and that this preference was consistent with higher ease, speed and accuracy scores. The only exception was in writing accuracy, where users tended to make more errors. However, most of these were attributed to a secondary characteristic of MAV-Vis (PoU coloring) which does not affect the semantic correctness of the model.

Our focus was notation design; in the future we intend to further investigate tooling integration, as well as the impact of tooling and user interface design on improving cognitive understanding. Another important future step is to determine whether the simple form of dependencies currently supported in MAV-Vis is sufficient. Finally, we intend to expand our investigation to include the case where elements are annotated with combinations of May, Var and Abs partiality, as well as to include OW partiality.

## REFERENCES

- [1] J. Aranda, N. Ernst, J. Horkoff, and S. Easterbrook. A framework for empirical evaluation of model comprehensibility. In *Proc. of MiSE'07*, page 7, 2007.
- [2] P. Chen. The Entity-Relationship Model: Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1:9–36, 1976.
- [3] M. Famelis. Uncertainty Management With Partial Models. In *Proc. of MODELS'12 Doc. Symp.*, 2012.
- [4] M. Famelis, M. Chechik, and R. Salay. Partial Models: Towards Modeling and Reasoning with Uncertainty. In *Proc. of ICSE'12*, pages 573–583, 2012.
- [5] S. Gopalakrishnan, J. Krogstie, and G. Sindre. Adapting UML Activity Diagrams for Mobile Work Process Modelling: Experimental Comparison of Two Notation Alternatives. In *Proc. of PoEM'10*, pages 145–161, 2010.
- [6] J.H. Larkin and H.A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11(1):65–100, 1987.
- [7] K. G. Larsen and B. Thomsen. A Modal Process Logic. In *Proc. of LICS'88*, pages 203–210, 1988.
- [8] D. Moody. The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *TSE*, 35(6):756–779, 2009.
- [9] D. Moody, P. Heymans, and R. Matulevičius. Visual syntax does matter: improving the cognitive effectiveness of the i\* visual notation. *REJ*, 15(2):141–175, 2010.
- [10] D. Moody and J. Hillegersberg. Evaluating the Visual Syntax of UML: An Analysis of the Cognitive Effectiveness of the UML Family of Diagrams. In *Proc. of SLE'09*, pages 16–34, 2009.
- [11] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann, 1994.
- [12] Object Management Group. *Meta Object Facility (MOF) Core Specification Version 2.0*, 2006.
- [13] OMG. *UML Superstructure Specification Version 2.3*, 2010.
- [14] R. Salay, M. Famelis, and M. Chechik. Language Independent Refinement using Partial Modeling. In *Proc. of FASE'12*, pages 224–239, April 2012.
- [15] R. Salay, J. Mylopoulos, and S. Easterbrook. Using Macromodels to Manage Collections of Related Models. In *Proc. of CaiSE'09*, pages 141–155, 2009.
- [16] P.Y. Schobbens, P. Heymans, and J.C. Trigaux. Feature diagrams: A survey and a formal semantics. In *Proc. of RE'06*, pages 139–148, 2006.