An Empirical Study Of Alternative Syntaxes For Expressing Model Uncertainty

Stephanie Santosa
University of Toronto
ssantosa@cs.toronto.edu

Michalis Famelis
University of Toronto
famelis@cs.toronto.edu

ABSTRACT

Partial models can be used to explicate design uncertainty in modeling artifacts. The existing, text-based syntax for partial models has been created in an ad-hoc manner without taking into account design guidelines for modeling notations. In this project, we perform an assessment of the effectiveness of the existing notation based on criteria outlined in the literature and propose a new, graphical notation following the same criteria. To evaluate both notations, we designed and executed an empirical user study to assess their cognitive effectiveness with regard to speed, ease and accuracy, as well as to get evidence about the preferences of users.

1. INTRODUCTION

In previous work [2], we have proposed *partial models* as a means to explicate and handle design uncertainty,i.e., uncertainty that the modeler may have about the content of her modeling artifacts. A partial model consists of a "base" model, decorated with *uncertainty annotations*, to express a set of possible conventional models. We refer to a specific decision about which the modeler is uncertain as a *Point of Uncertainty* (PoU). In [13], we introduced *MAVO partiality*, as a way to explicate uncertainty using syntactic annotations. We introduced four kinds of such annotations: (a) *May partiality*: annotating a model element with M indicates that we are unsure about whether it should exist in the model or not. *May* elements are also given a unique id element enclosed in a circle for use in a propositional formula indicating their groupings and dependencies. (b) *Abs partiality*: annotating an element with S indicates that we are unsure about whether it should actually be a collection of elements. (c) *Var partiality*: annotating an element with V indicates that we are unsure about whether it should actually be merged with other elements. (d) *OW partiality*: annotating the entire model with INC indicates that we are unsure about whether it is complete.

Example. We show an example partial Entity-Relational model in Figure 1. The model describes a hotel management system that consists of `Customer`s, that make `Reservation`s of `Room`s. `Customer`s, as well as hotel `Employee`s may have `Access` to `Room`s. The model contains three kinds of uncertainty May, Abs and Var. In particular, the model contains the following 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 whether she wants to associate `Access` with `Customer` in particular or `Person` in general.
- PoU3: The modeler does not know what `securityAttributes` the entity `Access` should have.
- PoU4: The modeler has not made up her mind about what entity should the property `internetAccess` be associated with and what the `id` attribute of such an entity-with-internet should be.

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.

Motivation and Contributions. Using partial models, we have shown how to model uncertainty [3], reason in its presence [4], systematically remove it using refinement [13], propagate it [14], manipulate and transform models that contain it [5], etc. These are ample evidence that partial models allow us to efficiently do *automated* reasoning in the presence of uncertainty. However, like all forms of modeling, apart from being formal, machine-processable artifacts, partial models are also a means of human communication. This means that partial models should be an effective means for *expressing* uncertainty to other people and for *understanding* the uncertainty expressed by other people. Both of these communication tasks are greatly informed by *notation* [8].

The partial model in Figure 1 is expressed in the notation introduced in [13], incorporating from [4] the notation for expressing May formulas. In particular, MAVO annotations are expressed textually and dependencies between PoUs are expressed in a propositional May Formula over a vocabulary of propositional variables, which are shown in the model using annotations in black circles. While partly influenced by work on behavioral modeling [7], This notation has been created in an ad-hoc manner, without taking into account the theory for the systematic design of notations proposed by D. Moody in [8]. We will refer to this notation as MAV-Text.

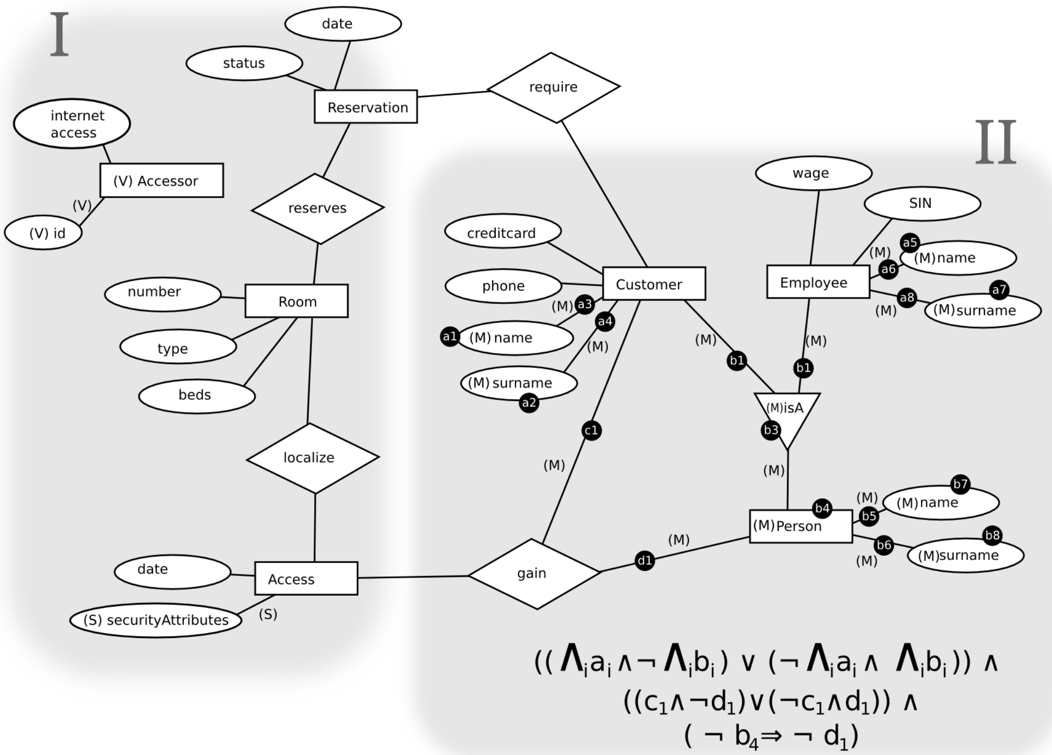

Figure 1: Motivating example in MAV-Text syntax. Region I: elements with Abs and Var uncertainty. Region II: elements May uncertainty and May formula.

Based on this observation, in this course project we attempted to do the following contributions:

C1: Provide an assessment of the existing state of partial modeling notation, using the criteria set forth in [8].

C2: Intentionally design a graphical notation, called MAV-Vis, for partial models that attempts to follow the same criteria.

C3: Empirically evaluate MAV-Text and MAV-Vis by conducting a user study to determine the effectiveness of each with respect to speed, ease and accuracy, (also defined in [8]) as well as to get evidence regarding user preferences.

Scope and Limitations. An important characteristic of MAV-Text and MAV-Vis is that they are annotation languages, as opposed to full-blown notations. In principle they could be used to annotate arbitrary models expressed in arbitrary languages. It is thus impossible to guarantee semiotic clarity (1:1 correspondence between symbols and concepts), because there can be no guarantee (from the designer of the annotation language) that the language of the annotated model will not use the same graphical constructs to indicate something else.

To address that concern we limit ourselves to the study of two specific modeling languages: Class Diagrams [12] and Entity-Relationship (E-R) Diagrams [1]. These choices are deliberate: (1) In MOF [11], Class Diagrams are used to express models of arbitrary metamodels in the abstract syntax. Therefore, even though MAV-Text and MAV-Vis may not be able to annotate arbitrary models in their concrete syntax, we can still annotate them when expressed in abstract syntax. (2) The E-R diagram is a simple, well-known notation taught in most undergraduate curricula. This should lower the language barrier to a level comparable to Class Diagrams.

To create MAV-Vis, we opted to forgo the use of propositional formulas like the one shown in Figure 1 for expressing dependencies between PoUs. Instead, we created means for users to express these dependencies graphically. I must be stressed that we are not attempting to create a graphical language for expressing arbitrary propositional expressions or even to express the most common propositional expression patterns encountered in partial modeling. We have deliberately kept the dependency notation simple as a first pass, while considering future scalability in the framework so more complexity can be added later. In Section 7, we briefly outline some ideas for further elaboration of the dependency notation.

Finally, in this project we have focused on May, Abs and Var because OW partiality is expressed at the model level. This means that the annotation language must be either be explicitly combined with tooling or expressed using languages that support multiple abstraction layers, for example macromodeling [15].

In Section 2, we present an analysis of MAV-Text, and in Section 3, we describe the new graphical notation for partial models, MAV-Vis. In Section 4 we give details about the empirical evaluation and in Section 5 we discuss and interpret our findings. We briefly discuss related work in Section 6 and conclude in Section 7.

2. ANALYSIS OF MAV-TEXT

In this section, we present our analysis of MAV-Text, based on the principles for designing effective visual notations outlined in [8]. In particular, we assessed the notation for Semiotic Clarity, Perceptual Discriminability, Semantic Transparency, Complexity Management, Cognitive Integration, Visual Expressiveness, Dual Coding, Graphic Economy and Cognitive Fit. Our results are summarized in the table shown in Figure 2. The components of MAV-Text- notations for May, Abs, and Var uncertainty is shown in the motivating example in Figure 1. The following is our assessment of these notations with respect to each principle.

Semiotic Clarity. *There should be a one-to-one correspondence between semantic constructs and graphical symbols.* This principle allows a notational system to constrain expression and interpretation without ambiguity. This avoids anomalies such as symbol redundancy, symbol overload, symbol excess, and symbol deficit.

As MAV uncertainties can apply to usage in different modeling languages, the notations selected to represent them must not overlap with those used by other modeling languages. We find that the annotation-based syntax satisfies the principle of semiotic clarity, as each type of uncertainty has a unique annotation associated with it, which is not shared with other semantic constructs in any of the other modeling languages considered. Note that here we consider different text annotations to be distinct, despite being homographs in terms of perceptual discriminability. Our treatment of this principle differs from other analyses [9], which would evaluate this as symbol overload. The use of the May formula is precise and expressive, and thus also satisfies this principle. In our example, there is no ambiguity in the meaning of the notations used, as each uncertainty is accurately notated.

Perceptual Discriminability. *Different symbols should be clearly distinguishable from each other.* This supports easy and accurate visual information processing. Discriminability of symbols is determined by the visual distance between them, measured by the number of visual variables on which they differ, and the magnitude of such variations. Shape is the primary visual variable for discriminability, and using more than one visual variable (redundant coding) to distinguish between symbols can further increase their visual distance.

The primary uncertainty type notations in MAV-Text are all text-based, and as such, have zero visual distance separating them according to this principle. Textual differentiation of symbols is cognitively inefficient for diagrams with high complexity, and interferes with the role of text as labels. We see in our example that (M), (V), and (S) annotations are not instantly discernable. Thus, this principle highlights an issue with MAV-Text annotations.

Semantic Transparency. *Use visual representations whose appearance suggests their meaning.* This relates to how intuitive and natural the symbol is for communicating the intended meaning. This also applies to relationships, where the spatial arrangements and connections between elements should also reflect how they are related.

This brings forth another potential issue with MAV-Text annotations, which are represented by the first letter of the name for the uncertainty concept they relate to. The lack of graphical symbol use is limiting to how intuitive the notation can be. We can see in our example that the (M), (S), and (V) annotations do not invoke their semantic meaning since they are not built upon any visual conventions for representing their corresponding uncertainties, so they are not likely not easily associated with their semantic concepts.

The May formula, however, relies on the established notation for propositional logic. The wide acceptance of this format may yield some "naturalness" in its use in MAV-Text for some users. Despite this, the relationships across different May elements (combinations and dependencies) are not visibly indicated on the diagram outside of the accompanying May formula.

Complexity Management. *Include explicit mechanisms for dealing with complexity.* Complexity is measured by the number of elements (symbol instances) on a diagram. Visual notations should contain mechanisms for managing complexity and avoiding cognitive overload. Such mechanisms can handle modularization and hierarchical structuring to break information into more manageable chunks and levels of abstraction.

This is a particularly relevant principle in the expression of uncertainty, as encoding non-concrete options into a partial model inherently adds complexity. MAV-Text introduces a new annotation element for each element with uncertainty, and the design for their use does not provide for any mechanisms for chunking information to manage this additional complexity. In our example, the simple may uncertainty alone introduces 34 new annotation elements. Multiple points of May uncertainty with dependencies across them can increase the number of alternatives to express in the May formula exponentially.

Cognitive Integration. *Include explicit mechanisms to support integration of information from different diagrams.* Representing systems across multiple diagrams requires the cognitive integration of these diagrams to form an understanding of the whole. Perceptual cues and contextualization techniques can indicate how the diagrams fit together and how to navigate between them.

MAV-Text does not provide specific mechanisms for cognitive integration, however, the May formula explicitly indicates how combinations of May elements as well as dependencies between them. This serves to contextualize individual may elements with respect to the overall uncertainty.

Visual Expressiveness. *Use the full range and capacities of visual variables. This measures the visual variation of the notation across the entire visual vocabulary.* The number of visual variables used to encode information determines the degree of visual expressiveness. Perceptually rich notations utilizing multiple visual channels provide higher degrees of expressiveness.

MAV-Text relies solely on textual encoding and thus has no information-carrying visual variables. While it is semantically expressive, this measures to zero-degrees of visual expressiveness. We observe in our example, that the only visual elements in MAV-Text are the circles enclosing May element labels. This principle suggests that exploiting more visual channels will yield a more cognitively effective notation.

Dual Coding. *Use text to complement graphics.* This principle is based on the dual coding theory, that text and graphics together can convey information more effectively than either on their own.

MAV-Text does not apply dual coding in any of its elements. Text is used for the encoding by itself, rather than as a complement to visual graphics. Additionally, the may for-

Principle	Rating	Assessment Summary
Semiotic Clarity	++	One-to-one correspondence to meaning
Perceptual Discriminability	--	Zero visual distance between notations
Semantic Transparency	-	Annotations not easily associated with concepts; Relationships not visible
Complexity Management	-	New annotation for each element with uncertainty No mechanisms for chunking information
Cognitive Integration		No specific mechanisms, but May formula contextualizes may elements to overall uncertainty
Visual Expressiveness	--	All textual encoding - measures to zero-degrees of visual expressiveness
Dual Coding	--	No dual coding; may formula is separated - spatial contiguity suggests in-place annotations
Graphic Economy	++	Not an issue - no use of graphic symbols
Cognitive Fit	+/-	Pen-and-paper appropriate. Requires a skill in propositional logic for may formula

Figure 2: Summary of our assessment of the MAV-Text syntax.

mula presents uncertainty alternatives separately from the diagram. In our example, it is displayed at the bottom of the diagram, with no visual connection to its referent elements. The principle of spatial contiguity suggests that in-place annotation would be more effective for this.

Graphic Economy. *The number of different graphical symbols should be cognitively manageable.* There are cognitive limits to the number of graphical symbols that can be effectively recognized. Too great a graphic complexity can place a burden on working memory, particularly if the symbols are not mnemonic and a legend is required.

This is not an issue with MAV-Text, but is something to note in further design of MAVO notation. Incorporating more types of uncertainty and developing constraints will require more expressiveness in the diagram.

Cognitive Fit. *Use different visual dialects for different tasks and audiences.* Cognitive fit in software modeling notation design is affected by user expertise and the representational medium available. This may result in complementary visual dialects to accommodate different users and mediums.

MAV-Text is appropriate for its representational medium, as it is suitable for both pen-and-paper as well as tooling solutions. However, it requires a certain amount of user skill in reading the may formula, which relies on knowledge of propositional logic. More complex notations, such as those used by the may formula in the example, can be problematic for users without the appropriate background. This is presents a limitation on accessibility.

3. DESIGN OF MAV-VIS

In this section, we briefly describe the MAV-Vis syntax for partial models and our design rationale for it. For comparison, the example of Figure 1 is shown in MAV-Vis in Figure 3.

We use color to distinguish between PoUs. In particular, each design decision about which the modeler is uncertain is given a unique color. In the example in Figure 3, 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 colored blue. The ability to distinguish between PoUs does not correspond directly to a concept in the formal, semantic domain of partial models (e.g. in MAV-Text there are no assumptions made about PoUs). We add this capability to MAV-Vis only to enhance understandability. Losing that information in e.g. a black-and-white printout does not remove essential information from the model.

Var uncertainty. Var uncertainty is represented by a *cloud* icon, as shown at the top of Region I in Figure 3. For node graphical elements and contained graphical elements (e.g. attributes in a class diagram), the icon annotation is placed to the left of the name label, as is done for the `Accessor` and `id` elements. For edge graphical elements, the annotation icon annotation is placed on top of the element, as is done for the edge element between `Accessor` and `id`. 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*. These are to be pen-and-paper friendly symbols that require minimal drawing skills so that the Cognitive Fit is appropriate.

Abs uncertainty. The representation of Abs uncertainty leverages a *pile* metaphor, as shown at the bottom of Region I in Figure 3. For node graphical elements, this is done by adding a second trace of the border of the node to look like a pile, as is done for the `securityAttributes` element. For contained gaphical elements (not shown in the example), the same effect is accomplished by first enclosing them in a box and then applying the same principle as for node elements. Edge graphical elements are annotated by adding a second line to give the appearance of "many lines", as is done for the edge between `SecurityClearance` and `securityAttributes`. We believe this would be an intuitive visual notation for the Set concept, improving Semantic Transparency as well as Visual Expressiveness and Perceptual Discriminatbility.

May uncertainty. In previous work, we have elaborated May uncertainty to also include a constraint language, namely propositional logic [4]. A model containing May-annotated

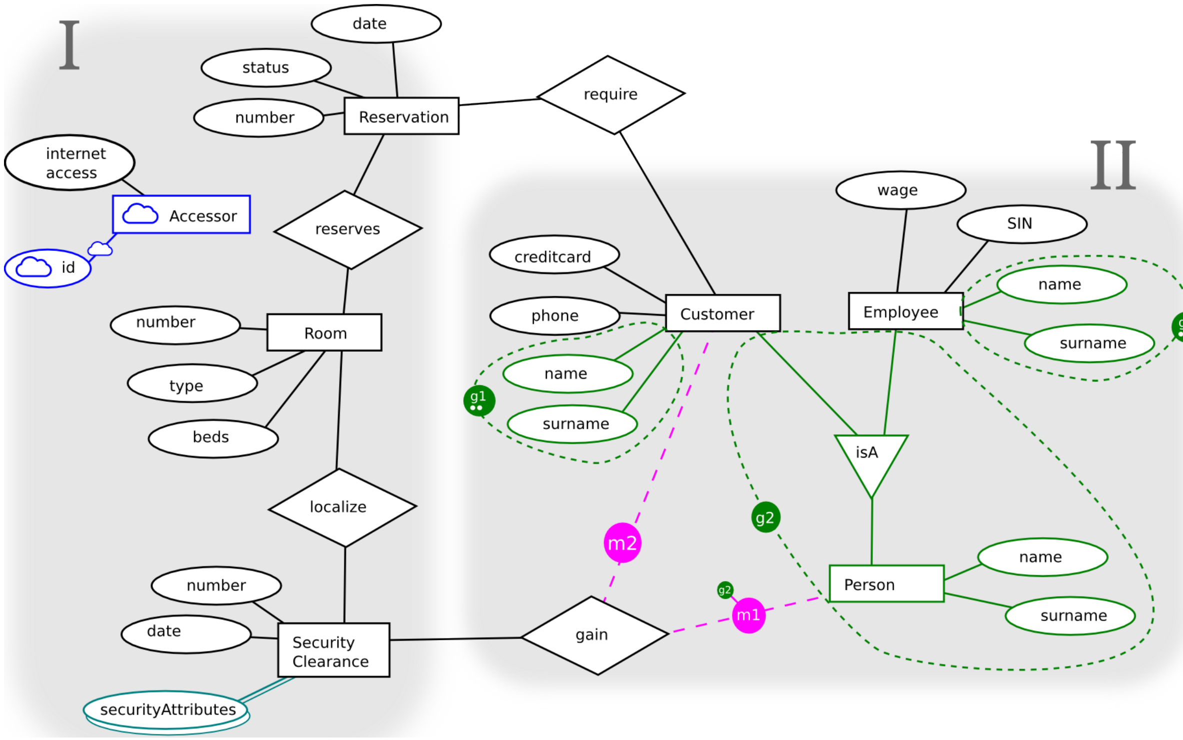

Figure 3: The model of Figure 1, expressed in MAV-Vis.

elements can also contain a propositional "May formula" which expresses the allowable configurations of May elements.

Explicating alternatives: 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 as follows: (a) All model elements that are part of the alternative are enclosed in a free-form dashed line. Alternatively, if A only has a single element, that element's border can be replaced by a dashed line. (b) If the elements making up A are spatially away from each other, each separate cluster is enclosed in a separate free-form dashed line. (c) The dashed line is annotated with a circular icon of the same color as the PoU for which A is an alternative. (d) The circular icon is also given a label of the form x_n where x is the name of the PoU and n is the ordinal number of the alternative. (e) If A consists on many clusters (cf. point (b)) then the the circular icon must contain a number of white dots equal to the number of clusters. (If there are more than 5 clusters, then instead of adding extra dots, the number of clusters is put inside a single dot.) The dots help the reader quickly identify that that the alternative has more than one cluster which she should locate elsewhere in the model.

The use of dashed line treatments for elements and enclosures here improves the Visual Expressiveness, adding the Texture visual variable to the notation and making the elements more easily distinguishable since they are identified with a separate visual variable form other notations. Additionally, the in-place grouping and identification scheme of the May elements exhibits Dual Coding with colour and prefix reflecting the point of uncertainty, and there is spatial contiguity between the May elements and their combinations.

We demonstrate these points in the example in Figure 3. In the example, PoU1 is represented by the color green and has two alternatives, g_1, g_2. The elements making up each alternative are enclosed by dashed lines. The alternative g_2 is contiguous, so its correposning circular icon only has the label $g2$, indicating that it is part of the "green" PoU and that it is the second alternative. The alternative g_1 comprises two separate clusters. The circular icon for each one has the appropriate label, as well as two white dots, to indicate that there are two clusters that make up the alternative.

Each element enclosed by a dashed line is assumed to be annotated with M. Assuming that a partial model M has k PoUs $\{P_1, ..., P_k\}$, and that a given PoU P_x has n alternatives $\{a_1^{P_x}, ..., a_n^{P_x}\}$ and that a given alternative a_y has l model elements, explicating May uncertainty as described above means that the May formula ϕ_M of M is:

$$\phi_M = \bigwedge_{x=1}^{k} \phi_{P_x}, \text{ where } \phi_{P_x} = \bigoplus_{y=1}^{n} \phi_{a_y}$$

where $\phi_{a_y} = \bigwedge_{z=1}^{l} e_z, e_z \in M \wedge e_z \in a_y$ (i.e., the conjunction of the propositional encoding of the elements in the alternative) and the symbol $\oplus$ denotes propositional exclusive-or (XOR).

Explicating dependencies between alternatives: MAV-Vis does not support explicating arbitrary dependencies between alternatives. Specifically, in this, preliminary version of the notation, it only supports a single, yet powerful type of dependency, namely, the case where the choice of one alternative in some PoU requires that some other alternative has been chosen at a different PoU. In our example in Figure 3, the modeler cannot choose the *dependee* alternative m_1 of

Principle	Rating	Assessment Summary
Semiotic Clarity	++	One-to-one correspondence to meaning
Perceptual Discriminability	++	Different retinal variables for each notation.
Semantic Transparency	+	Representations reflective of concepts; Relationships are visible
Complexity Management	++	Grouping applies uncertainty to entire submodels (not per element).
Cognitive Integration		No specific mechanisms, but May groupings and dot notation contextualize may elements to overall uncertainty
Visual Expressiveness	++	Visual expressiveness = 4. *Shape*: Icons and Piles, *Colour* for PoU's, *Texture*: Dashed line treatment, *Size*: may dependencies
Dual Coding	++	Color and text used together. In-place annotations for spatial contiguity.
Graphic Economy	+	High visual expressiveness keeps cognitively manageable (never exceeds 6 symbols per visual variable)
Cognitive Fit	+	No specialized skills required. Pen-and-paper appropriate.

Figure 4: Assessment of the MAV-Vis syntax (cf. Figure 2).

PoU2 (shown in magenta) unless she has also chose the *dependum* alternative g_2 of PoU1 (shown in green).

This kind of dependency is indicated by adding to the circular icon of the dependee a link to a small version of the icon of the dependum. In our example, the circular icon of m_1 is shown with a link to a small version of the icon of g_2. 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 $\{d_1, ..., d_N\}$. Explicating these relationships as described above means that for each dependee a_d the May formula ϕ_M is enhanced by conjuncting the expression:

$$\bigwedge_{x=1}^{N} (\neg d_x \Rightarrow \neg a_d)$$

We summarily present an evaluation of MAV-Vis based on the criteria outlined in [8] in the table shown in Figure 4.

4. EVALUATION

We conducted an experiment to evaluate and compare the cognitive effectiveness of our two notations: MAV-Text and MAV-Vis . Our goal is to address the following questions:

1. For each type of uncertainty, what is the cognitive effectiveness of reading and writing with each of the syntaxes?
2. What are the aspects that are most powerful and most problematic?
3. What notational syntax is preferred?

4.1 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 on items not clear in the tutorial. This task also provided insight into what types of notations people would naturally use to communicate these concepts. The participant was then given a reading and writing task using one syntax type, and this was 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). Figure 5 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 and Hotel Administration. The School Personnel model was a UML class diagram, while the Hotel Administration model was an E-R diagram.

We used a within-subjects design, to reduce selection bias and allow for each participant to compare notations and express their preferences. We control 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 measure cognitive effectiveness with respect to speed, ease and accuracy [8]. Speed is determined by task time, Ease is determined by questionnaire responses, and Accuracy is determined by writing error counts and comprehension scores. There are two components to accuracy that we are interested in observing in terms of the effect of the notation style: syntax correctness, and 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, participants were

Task ID	Name	Description	Uncertainty
A.	Free-Form (Write)	Simple base model given. Participant asked to add indicated points of uncertainty (PoU's).	3 PoU's: 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-Tex syntax. Participant asked to circle points of uncertainty, and indicate what the designer is uncertain about. Then asked to draw one example concretization for the Abs and Var PoU's, and all concretizations for the May PoU's.	Four PoU's: 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 PoU's 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 syntax. Participant asked to circle points of uncertainty, and indicate what the designer is uncertain about. Then asked to draw one example concretization for the Abs and Var PoU's, and all concretizations for the May PoU's.	Four PoU's: 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 PoU's are given: one Abs, one Var, and one May with 2 alternatives.

Figure 5: Modeling tasks of the evaluation.

given a summary sheet explaining first syntax type to read and use as reference for the corresponding tasks. Then, with one of the modeling scenarios described earlier, participants completed a reading task in that syntax type, followed by a writing task (B1-B2 or C1-C2). This was repeated with the other syntax type. At the end, participants filled out a post-study questionnaire where they were asked to rate both syntax for each uncertainty type, and indicate which syntax was preferred and why. Before and after all tasks in Table 5, participants recorded the displayed time.

Prior to running the experiment, a pilot evaluation was performed with one participant. Based on issues raised during this evaluation, clarifications were made to the instructions, and the syntax summary sheets were augmented with more detail.

Participants. 12 unpaid participants took part in the study. All participants had a Bachelors degree or higher in Computer Science and 9 were specialized in software engineering. The average experience level with MAVO uncertainty was 2.2 out of a 5-point Likert scale, however 3 were experts in MAVO uncertainty, and were already comfortable with MAV-Text. The average experience level with UML diagrams was 3.3, and the average experience level for E-R diagrams was 2.9.

Apparatus. The experiment was performed on pen-and-paper. Each participant was given 4 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. Participants could ask questions for clarification as needed throughout the experiment.

SPEED	Reading (mm:ss)	Writing (mm:ss)
MAV-Text	14:06	9:29
MAV-Vis	11:58	9:42

Figure 6: Speed results.

4.2 Results

Due to the limited number of participants, there was not enough data to perform statistical analysis. We report here on general observations on quantitative measures as well as qualitative feedback from participants.

Speed.

Table 6 summarizes the average completion times for each task. As expected, reading tasks took less time to complete using MAV-Vis, with MAV-Text averaging at 2:08 min longer to complete (17.8 percent). We also note that there was a substantial amount of overhead in these tasks, since the user was 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.

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 on a 5-point Likert scale, from 1 representing strong disagreement to 5 representing strong agreement. Therefore, 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 as a whole. Most participants preferred MAV-Vis overall. The tally of participants favoring MAV-Vis to MAV-Text was 8-2; with 2 participants indicating equal no preference between them.

Participants favoring MAV-Vis overall 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 participant indicated that *"annotations are very similar so meanings are almost overloaded. I had to think to remember the meanings"*. Two participants expressed that they would ideally like a mixed syntax, with Var and Abs following MAV-Vis notation, and May following MAV-Text notation. One of the neutral participants indicated preference for MAV-Text for writing and MAV-Vis for reading.

Abs: Table 7(a) shows the average scores for the Abs uncertainty type, along with the total count of participants preferring each syntax type used for representing it. Participants evaluated the MAV-Vis representation of the Abs

ABS	Intuitive	Easy to Remember	Efficient to Read	Efficient to Write	Number Preferred
MAV-Text	3.2	3.3	3.3	3.6	1
MAV-Vis	4.2	4.2	4.3	3.6	10

VAR	Intuitive	Easy to Remember	Efficient to Read	Efficient to Write	Number Preferred
MAV-Text	2.5	2.8	3.3	3.5	2
MAV-Vis	3.3	3.7	3.8	3.2	8

May	Intuitive	Easy to Remember	Efficient to Read	Efficient to Write	Number Preferred
MAV-Text	2.9	3.6	3.2	3.2	4
MAV-Vis	3.8	4.1	3.9	3.8	7

May Grouping	Intuitive	Easy to Remember	Efficient to Read	Efficient to Write	Number Preferred
MAV-Text	2.8	3.2	2.6	3.3	3
MAV-Vis	3.7	3.8	3.5	3.3	8

Figure 7: Ease results. (a) Abs (b) Var (c) May (d) May Groupings.

uncertainty to be more intuitive, easy to remember, and efficient to read, with a 1-point differential or more in each category. 10 participants preferred the MAV-Vis representation, versus 1 participant preferring the MAV-Text version.

Several participants preferring MAV-Vis commented that is was *"quick"* and *"easier to notice"*, *"intuitive"* and *"cognitively effective"*, making comments such as *"I could get it at the first glance"*. Other participants commented on the appropriateness of the representation; one stated that it was *"easy to associate with the concept"* while another stated that it *"clearly denoted that it was a collection"*.

The participant that preferred the MAV-Text representation found both notations equally intuitive, but indicated that *"(S) is easier to write than to draw a layered picture"*.

Var: Table 7(b) shows these results for the Var uncertainty type. While MAV-Vis was still the more favored syntax, with a MAV-Vis to MAV-Text preference tally of 8-2, the response to this representation was slightly less positive than with the Abs. Participants on average found it more intuitive, easy to remember, and efficient to read, with differences of 0.8, 0.9, and 0.5 points respectively. However they found it slightly less efficient to write by an average 0.3 points.

Participants were divided on the appropriateness of the symbol used. Some participants expressed that cloud notation was easily associated with the Var concept, while others had did not find it indicative of this, stating for example *"cloud does not equal var in my head"*. Two of the participants indicating this however, 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 gave for preferring for MAV-Text. The other participant preferring it stated that it takes longer to draw the cloud and *"takes up too much space"*.

May: The May uncertainty representation in MAV-Vis was rated higher than MAV-Text on average in all four categories, with margins of ranging from 0.5 to 0.9 across the categories. More participants preferred the MAV-Vis notation, but by a smaller margin, with 7 preferring MAV-Vis and 4 preferring MAV-Text. Table 7(c) displays these results.

READ	Abs (score/6)	Var (score/6)	May (score/6)	Total (score/18)
MAV-Text	3.9	5.1	2.8	11.8
MAV-Vis	3.9	5.2	4.2	13.3

WRITE	Syntax (error count)	Comprehension (error count)
MAV-Text	2.3	1.7
MAV-Vis	3.0	1.7

Figure 8: Accuracy results. (a) Reading scores (b) Writing error counts.

Many participants expressed the semantic clarity in using dashed lines for May elements, and indicated that it was easier to see and read, citing these as reasons they preferred MAV-Vis notation. Some of participants however, indicated that they found the (M) annotation to be cleaner, and easier to draw. These were the reasons backing their preference for MAV-Text in this case.

May Groupings: The MAV-Vis notation for May groupings were rated higher than its MAV-Text counterpart in the intuitive, easy to remember, and efficient to read categories, and in the efficient to write category they were rated equally at 3.3 points. MAV-Vis was also indicated as the preferred notation, with 8 participants favoring it, versus 3 favoring MAV-Text. Table 7(d) displays these results.

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"*. The participants that indicated a preference for MAV-Vis found the May formula *"more powerful"* and *"more commonly known"*.

Accuracy.

Reading: Reading performance for each type of uncertainty in was evaluated based on how successfully the 3 parts in the task was performed: Correct identification of Point of Uncertainty, correct identification of what the uncertainty is, and correct drawing of concretizations. Each of these parts was given 2 points for a correct solution, 1 point for a partially correct solution, and 0 points for an incorrect solution. Table 8(a) summarized the results for average score. There were similar performance levels for Abs and Var uncertainties across the two syntaxes, however the MAV-Vis syntax yielded much higher results for the May uncertainty and grouping, with an average score of 4.2 (versus 2.8 in MAV-Text).

Writing: Writing performance was evaluated as total error counts for syntax and comprehension across all uncertainty types. Table 8(b) displays the average error count in both categories for each of the syntaxes. We note that both syntaxes resulted in the same number of comprehension errors, but the number of syntax errors for MAV-Vis (averaging 3.0) was greater than those for MAV-Tex (averaging 2.3). However, an average of 1.7 of these errors in MAV-Vis were at-

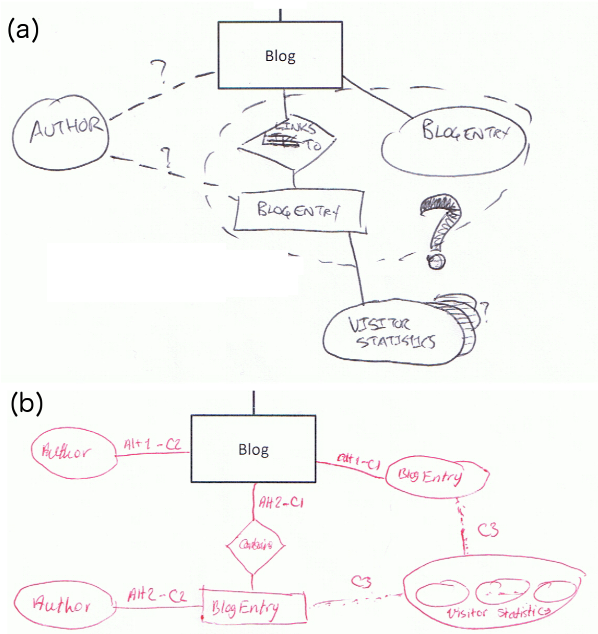

Figure 9: Examples of solutions given in the Free Form task.

tributable to incorrect/absent use of color-coding for grouping uncertainties.

Natural Ad-Hoc Notations. We looked at the notations invented in the Freehand task and observed some common elements across them. Dashed lines and questions marks were frequently used to express uncertainty of all types. Ellipses were also used in several examples to represent the Abs uncertainty. Additionally, we observed two different colour schemes: one colour per uncertainty type or a single colour different from the base model for all uncertain components. Figure 9 displays two examples of note-worthy solutions.

Threats to Validity. There are several factors that could have had some effect on the results. Having only 12 participants did not provide us with enough data to perform a statistical analysis, however it was sufficient to provide strong indicators on how the syntaxes compare. Another potential issue was that 3 participants reported extensive prior exposure to MAVO. This could have introduced some bias towards the MAV-Text syntax. Also, an imbalanced knowledge of UML vs E-R can result in some selection bias with our small sample size, and this was the case in one participant. This participant was in the control group with MAV-Text syntax in the more familiar base modeling language, so this also may have strengthened the MAV-Text results.

We also note that there were varying levels of proficiency with propositional logic amongst our participants. Since all were in the computer science field, this variation is likely to be reflective of the target audience for the syntax in any case. 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.

5. DISCUSSION

Our goals in the experiment were to measure the relative cognitive effectiveness of both syntaxes, identify their strengths, and weaknesses, and determine which was preferred. We now interpret the results for each uncertainty type with respect to these goals, and discuss the overall potential of the MAV-Vis syntax.

Abs Notation: We see that the pile metaphor was well-accepted and had good semantic transparency. While not introducing additional demand on the user for writing, the graphical notation in MAV-Vis improved the ease of use for writing, as indicated by participants in the questionnaire. While no difference was measured in accuracy, being much more intuitive and semantically connected to the Abs concept likely meant that the use of the pile notation was a factor in the reading tasks been a factor in the overall increased speed for reading tasks in MAV-Vis. Figure 9 (a) displays the notation by one participant in the Free-hand task, who independently came to a similar representation for Set.

Var Notation: Var uncertainty continues to be a difficult concept to form a notation for, due to its highly abstract nature. The polarized view on the cloud symbol in the MAV-Vis syntax shows room for improvement here. The fact that even participants who indicated this semantic disconnect preferred this notation over the (V) annotation due to its perceptual pop out. This indicates to us that the use of sketchable icons as uncertainty modifiers to existing elements is an appropriate solution. What the actual icon should be however is disputable. Participants also commented that name Var itself also did not yield automatic semantic association with the intended concept.

May and May Groupings Notation: As with the Abs Notation, the use of dashed lines in MAV-Vis associates naturally with its semantic concept. It is the only notation that ranks higher in all categories including writing. This is likely attributed to the perception of more efficiency in changing the line treatment or enclosing groups of items rather than creating separate notations for each May element.

While the margin of participants favouring MAV-Vis is not as high in this case, we note that 2 participants indicated their preference for the MAV-Text annotation was due to their familiarity with it from prior exposure. It is difficult to evaluate the representation of May with the May groupings completely separately, as their is overlap in representation of May and grouping May elements in the case when a dashed enclosure is used. We note that the other 2 participants indicating preference for the MAV-Text notation commented that it was due to their preference for the May formula.

There is a tradeoff with the MAV-Text and MAV-Vis notations between leveraging convention in using the propositional logic language that many computer scientists are already familiar with and introducing something new requiring some learning in exchange for added visualization power.

As an indicator that MAV-Vis is consistent with natural notations, we see that participants' notation in Figure 9 also used dashed lines to represent uncertainty and group elements with uncertainty, although in a different way. In (b) a May grouping scheme similar to MAV-Vis is applied, where the alternative number for each PoU is indicated in place on the diagram, but with annotations rather than visual elements.

Overall Syntax: Results for all uncertainty types favour MAV-Vis , indicating that the use of graphical elements in MAV-

Vis improves cognitive effectiveness, in particular for model reading. For writing, the syntaxes produced similar results, but very slightly favouring MAV-Text, particularly in the accuracy levels. Because the main differentiator in writing accuracy were the colour-coding errors, we can consider these results to be very similar as well. Colour coding is not a critical error, as it is most significant for use in May, and the diagram can be read without colouring points of uncertainty since the groupings are identified with prefix labels as well. Also, the notion and identification of points of uncertainty were added in MAV-Vis, and this information is not encapsulated by the MAV-Text syntax.

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. Though visual representation is indeed powerful, a more verbally-inclined learner may work better with the MAV-Text notation. 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. Additionally, MAV-Text notations are also more compatible for reasoning processes.

Tooling may be a good option for supporting both syntax styles and converting between them. Our focus was to develop a tool-independent pen-and-paper friendly notation, however this does not preclude the use of tooling to further enhance cognition in working with partial models. Additional interactions and visualization techniques can be developed to complement the syntax. For example, abstracting different levels of detail and supporting drill downs through them can improve cognitive integration, while hover highlights can display different concretizations for May alternatives.

6. RELATED WORK

Moody 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 [8]. This has been applied to asses the i* visual notation [9] and to evaluate the syntax of UML [10]. There has also been prior work in empirical syntax evaluation. An example of such is an experiment that was conducted to perform a comparison between two notational alternatives for process modeling [6]. We are also comparing two syntaxes empirically, although our metrics relate to the cognitive effectiveness criteria in [8].

7. CONCLUSION

In this project, we studied the syntax of notations that express uncertainty in models (partial modeling) based on the ideas presented by D.Moody in [8]. In particular, we did an assessment of the existing text-based notation, called MAV-Text and found that it suffers from serious design shortcomings. To address these, we constructed a visual notation, called MAV-Vis, using Moody's principles. We then designed and conducted a user-study, asking participants to perform a variety of modeling tasks using both notations.

Using the data generated by the case study, we were able to assess the two notations based on three notation effectiveness criteria, namely speed, easy and accuracy. 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 do more errors. However, most of these concerned a secondary characteristic of MAV-Vis, namely the different colouring of points of uncertainty, which does not affect the semantic correctness of the model and can be rectified by tooling.

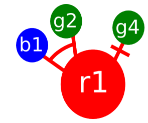

Figure 10: Mock-up: alternative r_1 from the "red" PoU requires either b_1 from the "blue" PoU or g_2 from the "green" PoU, while it is incompatible with g_4 from the "green" PoU.

In the future we intend to expand our investigation to include the case where elements are annotated with combinations of May, Var and Abs uncertainty, as well as to include OpenWorld uncertainty. We also intend to study the capabilities of MAV-Vis to express more complex dependencies between alternatives. Figure 10 shows a mock-up of one preliminary idea, which incorporates concepts from Feature Modeling in the syntax of the circular icon used to denote May uncertainty.

8. REFERENCES

[1] P. Chen. The Entity-Relationship Model: Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1:9–36, 1976.

[2] M. Famelis. Uncertainty Management With Partial Models. In *Proc. of MODELS'12 Doc. Symp.*, 2012.

[3] M. Famelis, Shoham Ben-David, Marsha Chechik, and Rick Salay. Partial Models: A Position Paper. In *Proc. of MoDeVVa'11*, pages 1–6, 2011.

[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] M. Famelis, R. Salay, and M. Chechik. The Semantics of Partial Model Transformations. In *Proc. of MiSE'12*, June 2012.

[6] 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.

[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] Object Management Group. *Meta Object Facility (MOF) Core Specification Version 2.0*, 2006.

[12] OMG. *UML Superstructure Specification Version 2.3*, 2010.

[13] R. Salay, M. Chechik, and J. Gorzny. Towards a Methodology for Verifying Partial Model Refinements. In *Proc. of VOLT'12*, pages 573–583, April 2012.

[14] R. Salay, M. Chechik, and J Horkoff. Managing Requirements Uncertainty with Partial Models. In *Proc. of RE'12*, September 2012.

[15] R. Salay, J. Mylopoulos, and S. Easterbrook. Using Macromodels to Manage Collections of Related Models. In *Proceedings of CaiSE'09*, pages 141–155, 2009.

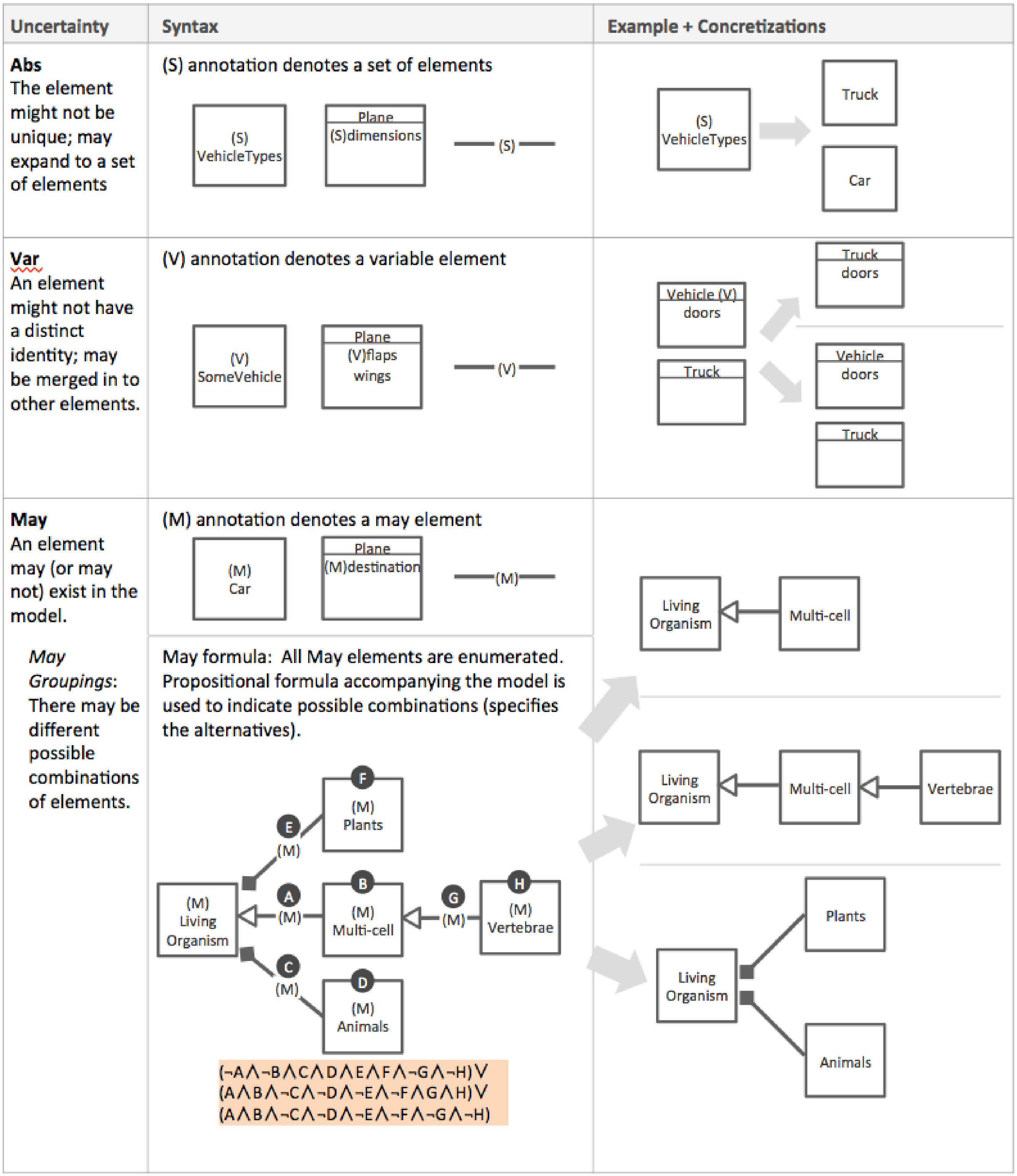

Figure 11: MAV-Text syntax summary.

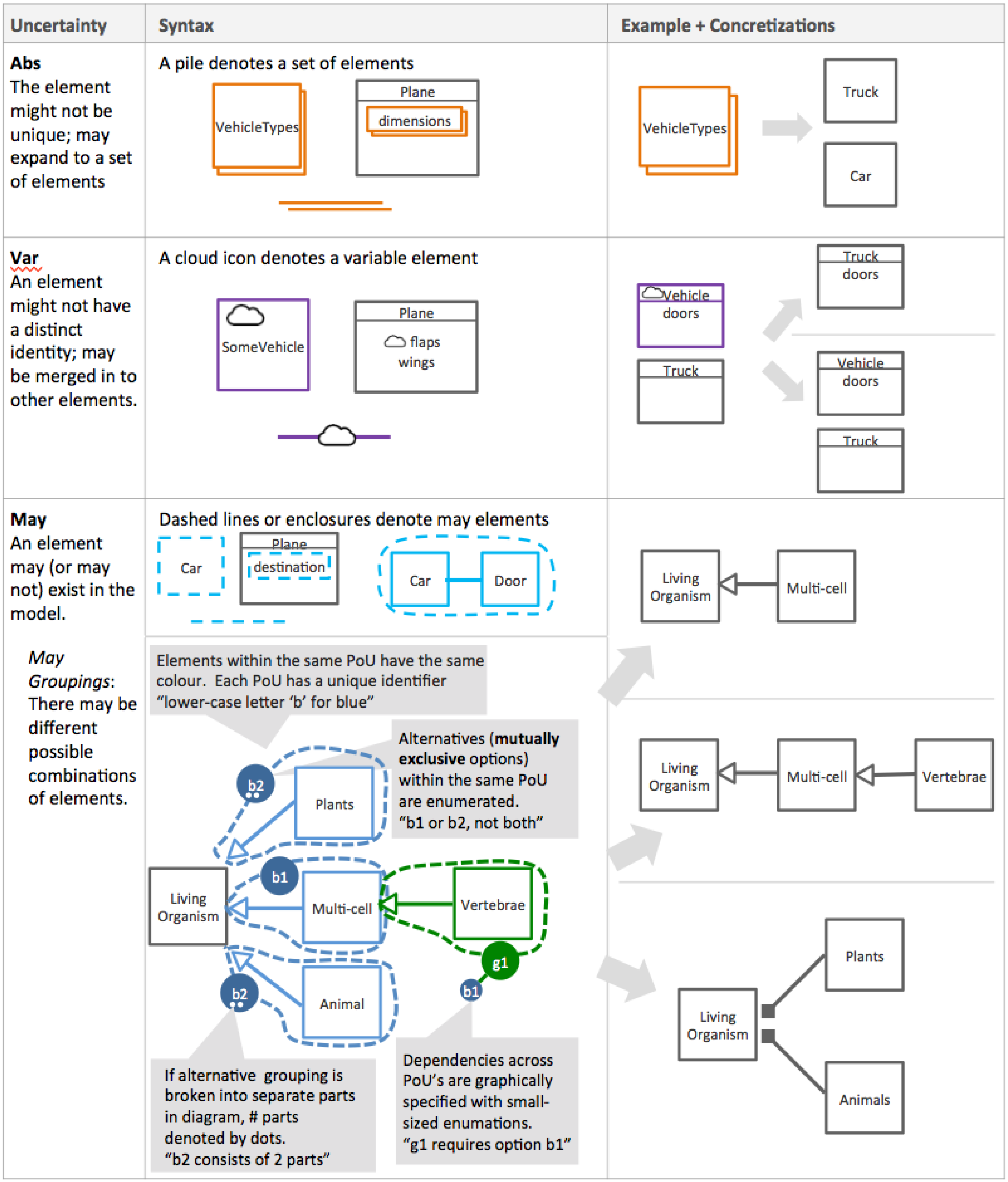

Figure 12: MAV-Vis syntax summary.